



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - K141502

# DETEKSI MOBIL DARI CITRA CCTV MENGUNAKAN FUZZY MORPHOLOGY DAN GRADIENT BOOSTING

ACHMAD SAIFUL  
NRP 5112 100 029

Dosen Pembimbing  
Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.  
Dr.Eng. Chastine Fatichah, S.Kom, M.Kom.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

*[Halaman ini sengaja dikosongkan]*



TUGAS AKHIR - K141502

# DETEKSI MOBIL DARI CITRA CCTV MENGUNAKAN FUZZY MORPHOLOGY DAN GRADIENT BOOSTING

ACHMAD SAIFUL  
NRP 5112 100 029

Dosen Pembimbing  
Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.  
Dr.Eng. Chastine Fatichah, S.Kom, M.Kom.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

*[Halaman ini sengaja dikosongkan]*



FINAL PROJECT - K141502

# VEHICLE DETECTION FROM CCTV IMAGES USING FUZZY MORPHOLOGY AND GRADIENT BOOSTING

ACHMAD SAIFUL  
NRP 5112 100 029

Advisor  
Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.  
Dr.Eng. Chastine Fatichah, S.Kom, M.Kom.

INFORMATICS DEPARTMENT  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### DETEKSI MOBIL DARI CITRA CCTV MENGUNAKAN FUZZY MORPHOLOGY DAN GRADIENT BOOSTING

#### TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Cerdas dan Visi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**ACHMAD SAIFUL**

NRP : 5112 100 029

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Prof. Dr. Ir. Joko Lianto Buliali, M.Kom .....  
NIP: 19670727 199203 1 002 ..... (pembimbing 1)

Dr.Eng. Chastine Fatichah, S.Kom, M.Kom .....  
NIP: 19751220 200112 2 001 ..... (pembimbing 2)



**SURABAYA  
MEI 2016**

*[Halaman ini sengaja dikosongkan]*



# **DETEKSI MOBIL DARI CITRA CCTV MENGUNAKAN FUZZY MORPHOLOGY DAN GRADIENT BOOSTING**

Nama Mahasiswa : Achmad Saiful  
NRP : 5112 100 029  
Jurusan : Teknik Informatika FTIF-ITS  
Dosen Pembimbing 1 : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.  
Dosen Pembimbing 2 : Dr.Eng. Chastine Fatichah, S.Kom,  
M.Kom.

## **ABSTRAK**

*Pendeteksian mobil memiliki peranan penting dalam manajemen sistem lalu lintas yang dinamis. Hal ini dikarenakan mobil lebih banyak menghabiskan badan jalan dibanding dengan kendaraan lain. Banyak metode yang dapat dilakukan untuk melakukan pendeteksian mobil, mulai dari menggunakan sensor hingga penggunaan kamera CCTV. Sebagai alat yang multifungsi dan lebih murah, pemerintah cenderung menerapkan kamera CCTV sebagai alat yang digunakan untuk memantau mobil di jalan raya maupun di jalan bebas hambatan. Oleh karena itu, diperlukan sebuah sistem deteksi mobil yang menggunakan citra CCTV yang akurat sehingga dapat digunakan untuk menetapkan durasi lampu lalu lintas.*

*Metode yang digunakan pada tugas akhir ini yaitu melakukan pendeteksian mobil dengan menggunakan tiga tahapan pada tahap pendeteksiannya. Yaitu pada tahap pertama yang akan dilakukan adalah tahap preprocessing. Pada tahap ini dilakukan operasi Median Filter, Resize Image, High Pass Filter untuk mendapatkan citra yang siap digunakan untuk tahapan selanjutnya. Tahap kedua yaitu tahapan processing. Pada tahapan ini digunakan operasi RGB to HSB, Fuzzy Dilation, Otsu Thresholding, Opening dan Connected Component untuk*

*mendapatkan kemungkinan mobil. Tahap ketiga yaitu tahap post-processing. Pada tahapan ini dilakukan ekstraksi fitur menggunakan Haar Wavelet dan Gradient Boosting untuk mendapatkan mobil. Hasil dari metode ini yaitu untuk nilai recall mencapai 65% dan untuk nilai precision yaitu 62%. Hal ini menandakan masih banyaknya mobil yang tidak terdeteksi dan false alarm yang cukup banyak.*

***Kata kunci: deteksi mobil, gradient boosting, operasi morfologi, pengolahan citra.***

# VEHICLE DETECTION FROM CCTV IMAGES USING FUZZY MORPHOLOGY AND GRADIENT BOOSTING

Name : Achmad Saiful  
NRP : 5112 100 029  
Major : Teknik Informatika FTIf-ITS  
Supervisor I : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.  
Supervisor II : Dr.Eng. Chastine Fatichah, S.Kom,  
M.Kom.

## ABSTRACT

*Car detection has an important role in dynamic traffic management systems. This is because car requires more space in the main road than other vehicles. Many methods can be done to make the detection of cars, ranging from the use of sensors to use CCTV cameras. As CCTV a multifunctional and cheaper tool, the government is likely to implement a CCTV camera as a tool used to monitor the cars on the main road or highway. Therefore, it is needed a car detection system which is very accurate so it can be used to set the duration of traffic lights.*

*The method used in this thesis are divided into three stage of detection phase. First stage is preprocessing stage. Median Filter, Resize Image, and High Pass Filter to get the image which be ready for the next stage will occur in this stage. Second stage is processing stage. Some operations are used in processing stage to get the possibility of the car which are RGM to HSB operation, Fuzzy Dilation, Otsu Thresholding, Opening and Connected Component. Third stage is post-processing stage. At this stage, feature extraction using Haar Wavelet and Gradient Boosting is done to get the car. The result show that the method can obtain recall value up to 65% and the for the precision value up to 62%. This result indicate that there are many cars are not detected and false alarms are quite a lot.*

***Keywords:*** *vehicle detection, gradient boosting, morphology operation, image processing.*

## **KATA PENGANTAR**

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **DETEKSI MOBIL DARI CITRA CCTV MENGGUNAKAN FUZZY MORFOLOGI DAN GRADIENT BOOSTING**

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, adik, kakak dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Prof. Dr. Ir. Joko Lianto Buliali, M.Sc. selaku dosen pembimbing tugas akhir pertama yang telah membimbing dan memberikan banyak masukan dalam pengerjaan tugas akhir ini.
3. Ibu Dr.Eng. Chastine Fatichah, S.Kom, M.Kom. selaku dosen pembimbing tugas akhir kedua yang selalu memberikan koreksi serta banyak masukan-masukan yang dapat penulis kembangkan dari tugas akhir ini
4. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak menyampaikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
5. Icing, Eric, Andrys, Nuzal, Raga yang selalu menemani, mengingatkan dan menyemangati dalam pengerjaan tugas akhir ini.
6. Niswah, Dita, Yoga, Zaki, Boim, dan teman teman SMA yang lain yang menemani penulis dalam suka maupun duka.
7. Teman teman BEM FTIf khususnya Departemen Sumber Daya Mahasiswa yang memberikan banyak pengalaman dalam berorganisasi.

8. Administrator Laboratorium Desain Terapan dan Komputasi dan administrator Laboratorium Komputasi Cerdas Visual yang memberikan kesempatan untuk mengerjakan tugas akhir di laboratorium.
9. Teman-teman angkatan 2012 Jurusan Teknik Informatika ITS, khususnya C1C yang telah menjadi teman seperjuangan dalam suka dan duka selama 4 tahun penulis menjalani kuliah.
10. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Mei 2016

Achmad Saiful

## DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxi
DAFTAR KODE SUMBER .....	xxiii
BAB I PENDAHULUAN .....	1
1.1.    Latar Belakang .....	1
1.2.    Tujuan.....	2
1.3.    Rumusan Permasalahan.....	2
1.4.    Batasan Permasalahan .....	2
1.5.    Metodologi .....	2
1.6.    Sistematika Penulisan.....	3
BAB II DASAR TEORI.....	7
2.1.    Median Filter .....	7
2.2.    High Pass Filter .....	7
2.3.    Dilation.....	8
2.4.    Eroton .....	9
2.5.    Fuzzy Morphology .....	9
2.6.    Otsu Thresholding .....	10
2.7.    Connected Component .....	10
2.8.    Haar Wavelet Transform .....	11
2.9.    Gradient Boosting.....	12
2.10.   Perhitungan Kinerja Aplikasi .....	13
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15
3.1.    Analisis Metode Secara Umum .....	15
3.2.    Perancangan Data .....	17
3.2.1.   Data Training.....	17
3.2.2.   Data Masukan.....	18
3.2.3.   Data Proses .....	18

3.2.4.	Data Keluaran .....	19
3.3.	Perancangan Proses .....	20
3.3.1.	Tahap Deteksi Kandidat Mobil .....	21
3.3.2.	Tahap Ekstraksi Fitur .....	25
3.3.3.	Tahap Training Data dan Klasifikasi .....	26
3.4.	Perancangan Antarmuka Perangkat Lunak .....	28
BAB IV IMPLEMENTASI .....		33
4.1.	Lingkungan Implementasi .....	33
4.2.	Implementasi Proses .....	33
4.2.1.	Implementasi Tahap Deteksi Kandidat Mobil .....	33
4.2.2.	Implementasi Tahap Ekstraksi Fitur .....	39
4.2.3.	Implementasi Training Data dan Klasifikasi .....	43
4.3.	Implementasi Antarmuka Perangkat Lunak .....	44
BAB V PENGUJIAN DAN EVALUASI .....		47
5.1.	Lingkungan Pengujian .....	47
5.2.	Data Uji Coba .....	47
5.3.	Skenario Uji Coba .....	47
5.4.	Skenario Pengujian 1: Perhitungan nilai <i>recall</i> , <i>precision</i> berdasarkan pengujian pada citra satu CCTV menggunakan Fuzzy Morphology. ....	48
5.5.	Skenario Pengujian 2: Perhitungan nilai <i>recall</i> , <i>precision</i> berdasarkan pengujian pada citra banyak CCTV menggunakan Fuzzy Morphology. ....	51
5.6.	Skenario Pengujian 3: Perhitungan nilai <i>recall</i> , <i>precision</i> berdasarkan pengujian pada citra satu CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel RGB. ....	51
5.7.	Skenario Pengujian 4: Perhitungan nilai <i>recall</i> , <i>precision</i> berdasarkan pengujian pada citra banyak CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel RGB. ....	54
5.8.	Skenario Pengujian 5: Perhitungan nilai <i>recall</i> , <i>precision</i> berdasarkan pengujian pada citra satu CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel Grayscale .....	55



5.9. Skenario Pengujian 6: Perhitungan nilai recall, precision berdasarkan pengujian pada citra banyak <i>CCTV</i> menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel Grayscale.....	58
5.10. Evaluasi .....	59
BAB VI KESIMPULAN DAN SARAN.....	63
6.1. Kesimpulan.....	63
6.2. Saran.....	64
DAFTAR PUSTAKA.....	65
Lampiran .....	67
BIODATA PENULIS.....	81

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Contoh Operasi <i>Median Filter</i> .....	7
Gambar 2.2 Contoh Operasi <i>High Pass Filter</i> .....	8
Gambar 2.3 Contoh Operasi <i>Dilation</i> .....	8
Gambar 2.4 Contoh Operasi <i>Eroton</i> .....	9
Gambar 2.5 Algoritma Tahap Pertama <i>Connected Component</i> ..	11
Gambar 2.6 Algoritma Tahap Kedua <i>Connected Component</i> .....	11
Gambar 2.7 <i>Wavelet Decomposition Levels</i> .....	12
Gambar 3.1 Gambar Diagram Alir Implementasi Metode Secara Umum 1 .....	16
Gambar 3.2 Gambar Diagram Alir Implementasi Metode Secara Umum 2 .....	17
Gambar 3.3 Contoh Sampel Negatif (i) dan Positif (ii).....	18
Gambar 3.4 Citra <i>CCTV</i> sebagai Data Masukan .....	19
Gambar 3.5 Hasil Keluaran .....	19
Gambar 3.6 Diagram Alir Tahap Deteksi Area Mobil .....	21
Gambar 3.7 Citra Hasil Operasi <i>Median Filter</i> .....	22
Gambar 3.8 Citra Hasil Operasi <i>High Pass Filter</i> .....	22
Gambar 3.9 Citra Hasil <i>Saturation Channel</i> .....	23
Gambar 3.10 Citra Hasil <i>Fuzzy Dilation</i> .....	23
Gambar 3.11 Citra Hasil Operasi <i>Otsu Thresholding</i> .....	24
Gambar 3.12 Hasil dari Operasi <i>Opening</i> Dan <i>Eroton</i> .....	25
Gambar 3.13 Rancangan Halaman Pertama .....	28
Gambar 3.14 <i>Widget File Chooser</i> .....	28
Gambar 3.15 Rancangan Halaman Kedua.....	29
Gambar 3.16 Rancangan Halaman Ketiga .....	30
Gambar 4.1 Implementasi Halaman Terakhir .....	45
Gambar 5.1 Hasil Pengujian pada Citra <i>CCTV</i> Semarang Menggunakan Fuzzy Morphology .....	49
Gambar 5.2 Hasil Pengujian pada Citra <i>CCTV</i> Cipularang Menggunakan Fuzzy Morphology .....	50
Gambar 5.3 Hasil Pengujian pada Citra <i>CCTV</i> Padalarang Menggunakan Fuzzy Morphology .....	50

Gambar 5.4 Pengujian pada Citra Banyak CCTV Menggunakan Fuzzy Morphology .....	51
Gambar 5.5 Hasil Pengujian pada Citra CCTV Semarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB .....	53
Gambar 5.6 Hasil Pengujian pada Citra CCTV Cipularang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB .....	53
Gambar 5.7 Hasil Pengujian pada Citra CCTV Padalarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB .....	54
Gambar 5.8 Hasil Pengujian pada Citra Banyak CCTV Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB .....	55
Gambar 5.9 Hasil Pengujian pada Citra CCTV Semarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale .....	56
Gambar 5.10 Hasil Pengujian pada Citra CCTV Cipularang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale. ....	57
Gambar 5.11 Hasil Pengujian pada Citra CCTV Padalarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale .....	57
Gambar 5.12 Hasil Pengujian pada Citra Banyak CCTV Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale .....	58
Gambar 5.13 Hasil Uji Coba Menggunakan Fuzzy Morphology pada CCTV Padalarang (i), CCTV Cipularang (ii), dan CCTV Semarang (iii) .....	59
Gambar 5.14 Hasil Uji Coba Menggunakan Fuzzy Morphology pada Banyak CCTV dengan Parameter yang Sama .....	59
Gambar 5.15 Citra CCTV dengan Kendaraan yang Berdekatan..	61

## **DAFTAR TABEL**

Tabel 3.1 Data Proses .....	20
Tabel 3.2 Spesifikasi atribut antarmuka halaman pertama .....	28
Tabel 3.3 Spesifikasi Atribut antarmuka Halaman Kedua .....	30
Tabel 3.4 Spesifikasi Atribut antarmuka Halaman Ketiga .....	31
Tabel 4.1 Lingkungan Implementasi Sistem .....	33
Tabel 5.1 Perbandingan Nilai Recall dan Precision Hasil Fuzzy Morphology antara Satu CCTV dengan Banyak CCTV .....	59
Tabel 5.2 Perbandingan Nilai Recall dan Precision pada Penghitungan Mobil pada Satu CCTV Menggunakan Ekstraksi Fitur Channel Grayscale dengan Channel RGB .....	60
Tabel 5.3 Perbandingan Nilai Recall dan Precision pada Penghitungan Mobil pada Banyak CCTV Menggunakan Ekstraksi Fitur Channel Grayscale dengan Channel RGB .....	60

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode Program <i>Median Filter Channel Merah</i>	34
Kode Sumber 4.2 Kode Program <i>High Pass Filter Channel Merah</i>	35
Kode Sumber 4.3 Kode Program untuk Mendapatkan <i>Channel Saturation</i>	35
Kode Sumber 4.4 Kode Program Fungsi Utama Fuzzy Dilation	36
Kode Sumber 4.5 Kode Program Fuzzy untuk Menentukan Tingkat Keabuan	36
Kode Sumber 4.6 Kode Program <i>Otsu</i> untuk Mendapatkan Nilai <i>Threshold</i>	37
Kode Sumber 4.7 Kode Program <i>Opening</i>	38
Kode Sumber 4.8 Kode untuk Mendapatkan Nilai Intensitas <i>Erosion</i>	38
Kode Sumber 4.9 Kode untuk Mendapatkan Nilai Intensitas <i>Dilation</i>	39
Kode Sumber 4.10 Fungsi <i>Connected Component</i>	39
Kode Sumber 4.11 Kode Program <i>Haar Wavelet Transform Channel Merah</i>	40
Kode Sumber 4.12 Kode Program Ekstraksi Fitur pada <i>Channel Merah</i>	41
Kode Sumber 4.13 Kode Program Normalisasi Fitur	42
Kode Sumber 4.14 Implementasi Gradient Boosting (I)	43
Kode Sumber 4.15 Implementasi <i>Gradient Boosting (II)</i>	44
Kode Sumber 4.16 Perintah pada Tombol Proses	44

*[Halaman ini sengaja dikosongkan]*



# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### **1.1. Latar Belakang**

Saat ini, kota kota besar di Indonesia memiliki masalah yang sama, yaitu kemacetan. Kemacetan timbul akibat banyaknya jumlah kendaraan daripada kapasitas jalan raya yang ada. Membuka jalan raya baru merupakan hal yang dapat menyelesaikan masalah tersebut, tetapi hal itu sulit untuk direalisasikan karena lahan di kota kota besar sudah habis untuk bangunan, sehingga tidak memungkinkan untuk dibukanya jalan raya baru. Salah satu cara untuk memecahkan masalah tersebut adalah dengan sistem manajemen lalu lintas. Sistem manajemen lalu lintas berfungsi untuk mengatur arus lalu lintas yang ada di jalan raya. Salah satu sistem manajemen lalu lintas adalah pengendalian arus pada persimpangan jalan menggunakan lampu lalu lintas, dengan lampu lalu lintas kemacetan pada jalan raya dapat teratasi. Durasi lampu lalu lintas dapat ditentukan berdasarkan jumlah kendaraan pada setiap sisi persimpangan.

Pada tugas akhir ini dilakukan penghitungan mobil pada pagi hari dengan kondisi cuaca cerah, penghitungan ini menggunakan tiga tahapan, tahap yang pertama yaitu *pre-processing* yaitu tahapan meningkatkan kualitas citra menggunakan *Median Filter*, *Resize Image*, dan *High Pass Filter*. Tahap kedua yaitu *processing* yaitu tahapan untuk memisahkan objek dan *background* menggunakan *RGB to HSB*, *Fuzzy Dilation*, *Otsu Thresholding*, *Opening*, dan *Connected Component*. Tahap ketiga yaitu tahap *post-processing* yaitu tahapan mengenali objek sebagai mobil atau bukan mobil, tahapan ini menggunakan *Haar Wavelet* dan *Gradient Boosting*.

Hasil dari aplikasi ini diharapkan dapat menjadi bahan pertimbangan untuk manajemen lalu lintas dengan cara pengaturan durasi lampu merah.

## **1.2. Tujuan**

Tujuan dari pembuatan tugas akhir ini yaitu membuat program yang dapat menghitung banyaknya kendaraan pada gambar *CCTV* dengan melakukan operasi *fuzzy morphology* dan juga menggunakan *Gradient Boosting* sebagai *machine learning* untuk membangun *classifier*. Program dibangun menggunakan pemrograman Java berbasis *desktop*.

## **1.3. Rumusan Permasalahan**

Rumusan masalah yang diangkat dalam tugas akhir ini antara lain:

1. Bagaimana cara mendeteksi kendaraan pada pagi hari.
2. Bagaimana cara membedakan antara mobil dengan objek yang lain.
3. Bagaimana cara menerapkan pengolahan citra pada pemrograman java.

## **1.4. Batasan Permasalahan**

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, antara lain:

1. Gambar yang digunakan memiliki ukuran maksimal 400 x 400 pixel
2. Gambar yang digunakan yaitu gambar *CCTV* pada pagi hari dengan kondisi cuaca cerah.
3. Kendaraan yang dideteksi hanya mobil.

## **1.5. Metodologi**

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

### **1. Studi literatur**

Pada studi literatur ini, akan dipelajari sejumlah

referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai Java (software platform), operasi Morfologi citra, *Haar Wavelet*, dan *Gradient Boosting* sebagai *machine learning*.

## **2. Implementasi**

Dalam tugas akhir ini nantinya akan menghasilkan jumlah kendaraan pada gambar *CCTV*. Masukan data berupa gambar *CCTV* jalan raya pada malam hari. Hasil perangkat lunak berupa gambar dengan label kendaraan (bukan sepeda motor) dalam keadaan malam hari. Berikut beberapa hal yang diperlukan dalam implementasi:

- a. IDE Netbeans 8.0.2
- b. Kemampuan Bahasa Pemrograman Java
- c. Java Development Kit 8.
- d. Apache Spark

## **3. Pengujian dan evaluasi**

Pengujian dari aplikasi ini akan dilakukan dengan cara pengujian blackbox. Pengujian blackbox adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Pengujian ini dilakukan untuk menguji apakah posting terkelompokkan ke dalam kategori yang sesuai atau tidak.

## **4. Penyusunan buku Tugas Akhir**

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

### **1.6. Sistematika Penulisan**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan

pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

**Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

**Bab II Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

**Bab III Metode Pemecahan Masalah**

Bab ini membahas mengenai Metode yang digunakan untuk memecahkan masalah yang dipaparkan pada rumusan permasalahan.

**Bab IV Analisis dan Perancangan Sistem**

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

**Bab V Implementasi**

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

**Bab VI Pengujian dan Evaluasi**

Bab ini membahas pengujian dengan metode kotak hitam (*black box testing*) untuk mengetahui aspek nilai fungsionalitas dari perangkat lunak dan nilai kegunaan yang dibuat dengan juga memperhatikan ketertarikan pada calon partisipan untuk menggunakan aplikasi ini.

**Bab VII Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

**Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

**Lampiran**

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

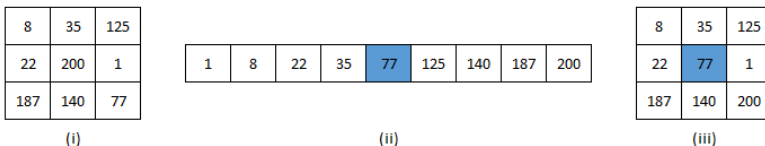
*[Halaman ini sengaja dikosongkan]*

## BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Teori-teori tersebut meliputi pengertian dan beberapa analisis terkait pendeteksian mobil pada malam hari.

### 2.1. Median Filter

*Median filter* adalah salah satu *filtering* non-linear yang mengurutkan nilai intensitas sekelompok *pixel*, kemudian mengganti nilai *pixel* yang diproses dengan nilai mediannya [1]. *Median filter* berguna untuk menghilangkan *salt and pepper noise*.



**Gambar 2.1 Contoh Operasi *Median Filter***

Pada Gambar 2.1 Contoh Operasi *Median Filter* bagian (i) merupakan nilai intensitas yang akan di proses dengan *structuring element* 3x3, bagian (ii) merupakan operasi median filter dimana seluruh pixel diurutkan mulai dari nilai terkecil ke terbesar kemudian nilai tengah sebagai nilai akhir, bagian (iii) merupakan nilai akhir, nilai 200 diganti dengan 77.

### 2.2. High Pass Filter

*High pass filter* merupakan dasar dari semua metode penajaman gambar. *High pass filter* adalah suatu proses pada gambar/citra dari bentuk filter yang mengambil komponen frekuensi tinggi dan menurunkan komponen frekuensi rendah [2]. Operasi high pass filter ini mengalikan nilai intensitas sekelompok pixel dengan kernel filter yang telah ditentukan.

8	35	125
22	60	1
187	140	77

(i)

0	-1	0
-1	6	-1
0	-1	0

(ii)

8	35	125
22	162	1
187	140	200

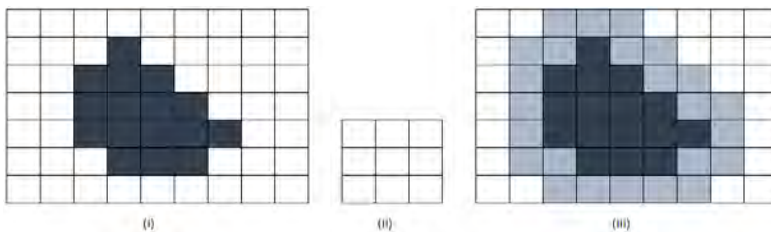
(iii)

**Gambar 2.2 Contoh Operasi *High Pass Filter***

Pada Gambar 2.2 Contoh Operasi *High Pass Filter*, bagian (i) merupakan nilai intensitas yang akan di proses dengan *structuring element* 3x3, bagian (ii) merupakan kernel *high pass filter*, bagian (iii) merupakan hasil penjumlahan dari perkalian bagian (i) dengan bagian (ii) tiap pixel dengan index yang sama.

### 2.3. Dilation

*Dilation* adalah suatu proses menambahkan piksel pada batasan objek dalam suatu citra sehingga apabila dilakukan operasi maka hasilnya akan menjadi lebih besar dibandingkan citra aslinya [3].



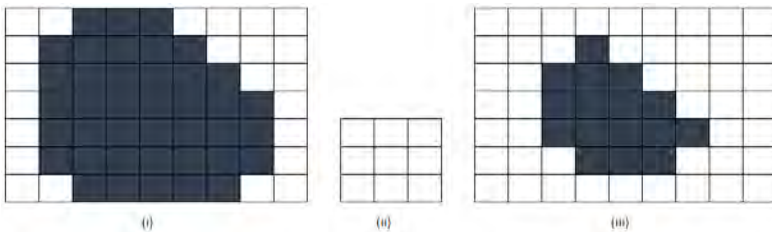
**Gambar 2.3 Contoh Operasi *Dilation***

Pada Gambar 2.3 Contoh Operasi *Dilation*, bagian (i) merupakan citra masukan, bagian (ii) merupakan *structuring element* 3x3, bagian (iii) merupakan citra setelah dilakukan dilation. Operasi yang dilakukan pada Gambar 2.3 Contoh Operasi *Dilation* adalah mengubah nilai 0 disekitar piksel yang dipilih seluas *structuring element* menjadi nilai 1.



## 2.4. Erosion

*Erosion* adalah suatu proses mengurangi piksel pada batasan objek dalam suatu citra sehingga apabila dilakukan operasi maka hasilnya akan menjadi lebih kecil dibandingkan citra aslinya [3].



**Gambar 2.4 Contoh Operasi *Erosion***

Pada Gambar 2.4 Contoh Operasi *Erosion*, bagian (i) merupakan citra masukan, bagian (ii) merupakan *structuring element* 3x3, bagian (iii) merupakan citra setelah dilakukan *erosion*. Operasi yang dilakukan adalah mengubah nilai piksel yang dipilih menjadi 0 apabila disekitar piksel yang dipilih seluas *structuring element* terdapat nilai 0.

## 2.5. Fuzzy Morphology

Logika fuzzy adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian. Dimana logika klasik (crisp) menyatakan bahwa segala hal dapat diekspresikan dalam istilah binary (0 atau 1, hitam atau putih, ya atau tidak). Logika fuzzy menggantikan kebenaran Boolean dengan tingkat kebenaran. Logika fuzzy memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistic, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat".

Operasi morfologi adalah teknik pengolahan citra yang didasarkan pada bentuk segmen atau region dalam citra. Karena difokuskan pada bentuk obyek, maka operasi ini biasanya diterapkan pada citra biner. Biasanya segmen tadi didasarkan pada obyek yang menjadi perhatian. Segmentasi dilakukan dengan

membedakan antara obyek dan latar, antara lain dengan memanfaatkan operasi pengambangan yang mengubah citra warna dan skala keabuan menjadi citra biner.

*Fuzzy Morphology* mengadopsi pendekatan logika fuzzy untuk *grayscale morphology* dan mengkombinasikannya dengan konsep adjunctions [4]. Konsep logika fuzzy untuk memperpanjang morfologi biner menjadi *graylevel images*. Logika fuzzy digunakan pada operasi morfologi dilasi dan erosi. *Fuzzy erosion* dari gambar  $f$  dapat didefinisikan dengan struktur  $B$  di titik  $x$ . Ditunjukkan dengan persamaan (2.1). *fuzzy dilation* dari gambar  $f$  dapat didefinisikan dengan struktur  $B$  di titik  $x$  ditunjukkan dengan persamaan (2.2) dan juga tahapan teori morfologi, fuzzy opening (2.3) dan persamaan fuzzy closing (2.4)

$$\varepsilon^F(f, B)(x) := \inf_{y \in f} \{I(B(y), f(y))\} \quad (2.1)$$

$$\delta^F(f, B)(x) := \sup_{y \in f} \{I(B(y), f(y))\} \quad (2.2)$$

$$\gamma^F(f, B)(x) := \delta^F(\varepsilon^F(f, B), B) \quad (2.3)$$

$$\Phi^F(f, B)(x) := \varepsilon^F(\delta^F(f, B), B) \quad (2.4)$$

## 2.6. Otsu Thresholding

*Otsu Thresholding* adalah mengubah citra berderajat keabuan menjadi citra biner atau hitam putih secara otomatis tanpa memasukkan nilai ambang. Nilai ambang batas didapat dari metode otsu [5]. pendekatan yang dilakukan oleh metode otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variable yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimumkan variable agar dapat membagi objek *background* dan *foreground*.

## 2.7. Connected Component

*Connected Component* adalah menggabungkan beberapa pixel yang sama dan berdekatan menjadi objek, dalam satu citra akan didapatkan banyak objek. Pada umumnya operasi ini

dilakukan pada citra biner, sehingga dapat lebih mudah untuk mendeteksi objek objek.

*Connected Component* bekerja dengan cara memindai gambar, dari piksel atas kiri sampai piksel bawah kanan untuk mengidentifikasi piksel yang terhubung [6]. Piksel yang terhubung adalah piksel yang berdekatan dan mempunyai intensitas yang sama. Operasi ini dilakukan pada citra biner atau citra dengan derajat keabuan dan citra dengan intensitas lain yang memungkinkan.

*Connected Component* melakukan dua kali tahapan pada citra, tahap pertama dapat dilihat pada Gambar 2.5 yaitu memberikan label sementara pada tiap pixel. tahap kedua dapat dilihat pada Gambar 2.6 yaitu mengganti label sementara dengan label terkecil.

1	Iterasi setiap piksel pada seluruh gambar
2	If piksel bukan background
3	Get label piksel disekitarnya
4	If nilai piksel disekitarnya belum ada label
5	Set label baru
6	Else
7	Min = Get label terkecil disekitarnya
8	label piksel = Min

**Gambar 2.5 Algoritma Tahap Pertama *Connected Component***

1	Iterasi setiap piksel pada seluruh gambar
2	If piksel bukan background
3	Set label dengan label terkecil

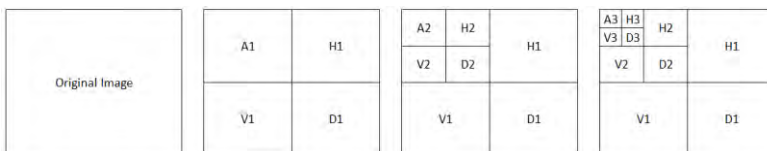
**Gambar 2.6 Algoritma Tahap Kedua *Connected Component***

## 2.8. Haar Wavelet Transform

Wavelet adalah fungsi matematika yang digunakan untuk membagi suatu fungsi atau sinyal waktu kontinu ke dalam komponen skala yang berbeda. Dalam proses skala biasanya dapat menetapkan rentang frekuensi untuk setiap skalanya [7]. Wavelet transform merepresentasikan fungsi wavelet. Wavelet membagi

data menjadi komponen frekuensi yang berbeda, kemudian mempelajari setiap komponen dengan resolusi yang telah disamakan dengan skalanya. keunggulan dibanding dengan metode fourier tradisional dalam menganalisa situasi fisik dimana sinyal berisi diskontinuitas dan lonjakan tajam. jika objek kecil dalam ukuran atau rendah dalam contrast, diuji dalam resolusi tinggi.

Wavelet transform dihitung secara terpisah untuk segmen yang berbeda dari sinyal domain waktu pada frekuensi yang berbeda. Transformasi wavelet menguraikan sinyal menjadi empat rinci yaitu rincian perkiraan, rincian horisontal, rincian vertikal dan rincian diagonal seperti ditunjukkan pada Gambar 2.7. Dengan menggunakan transformasi wavelet didapatkan informasi lengkap dari gambar di semua frekuensi.



**Gambar 2.7 Wavelet Decomposition Levels**

## 2.9. Gradient Boosting

Meta-algoritma adalah algoritma yang menggunakan algoritma lain sebagai perwakilan, dan juga merupakan algoritma yang memiliki sub-algoritma sebagai peubah dan parameter yang dapat diganti. Contoh-contoh yang termasuk meta-algoritma adalah *Boosting*.

*Boosting* merupakan meta-algoritma dalam *machine learning* untuk melakukan *supervised learning*. Secara umum, boosting terjadi dalam iterasi, secara *incremental* menambahkan *weak learner* ke dalam satu *strong learner*. Pada setiap iterasi, satu *weak learner* belajar dari suatu data latihan. Kemudian, *weak learner* itu ditambahkan ke dalam *strong learner*. Setelah *weak learner* ditambahkan, data-data kemudian diubah masing-masing bobotnya. Data-data yang mengalami kesalahan klasifikasi akan mengalami penambahan bobot, dan data-data yang terklasifikasi

dengan benar akan mengalami pengurangan bobot [8]. Oleh karena itu, *weak learner* pada iterasi selanjutnya akan lebih terfokus pada data-data yang mengalami kesalahan klasifikasi oleh *weak learner* yang sebelumnya.

*Gradient boosting* adalah salah satu dari algoritma boosting, dimana menghasilkan model prediksi dari *weak learner* berbentuk *decision tree*. *Gradient boosting* melatih *decision tree* untuk meminimalkan *loss function*, *Gradient boosting* menangani fitur kategoris, perluasan dari *multiclass classification setting*, tidak memerlukan fitur *scaling*, dan dapat menangkap fitur non-linearities dan interaksi [9].

## 2.10. Perhitungan Kinerja Aplikasi

Perhitungan kinerja aplikasi pada sistem ini adalah dengan menggunakan akurasi. Akurasi digunakan untuk mengukur tingkat kualitas deteksi mobil. Kategori uji coba dinyatakan benar jika hasil deteksi adalah mobil. Sedangkan kategori uji coba dinyatakan salah jika hasil deteksi bukanlah objek mobil. Perhitungan akurasi menggunakan *Precision* and *Recall* [10]. Persamaan untuk pengukuran akurasi dapat dilihat pada persamaan 2.5 dan persamaan 2.6:

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \quad (2.5)$$

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (2.6)$$

Pada persamaan diatas, *true positive* pada pengujian ini adalah jumlah mobil yang ada pada citra, *false negative* adalah jumlah kesalahan deteksi yang mobil tapi tidak terdeteksi (*missing rate*), dan *false positive* adalah jumlah kesalahan deteksi yang bukan mobil tapi terdeteksi mobil.

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Pada Bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari tugas akhir. Perancangan ini meliputi perancangan data, perancangan proses, dan perancangan antar muka, serta juga akan dijelaskan tentang analisis implementasi metode secara umum pada sistem.

#### **3.1. Analisis Metode Secara Umum**

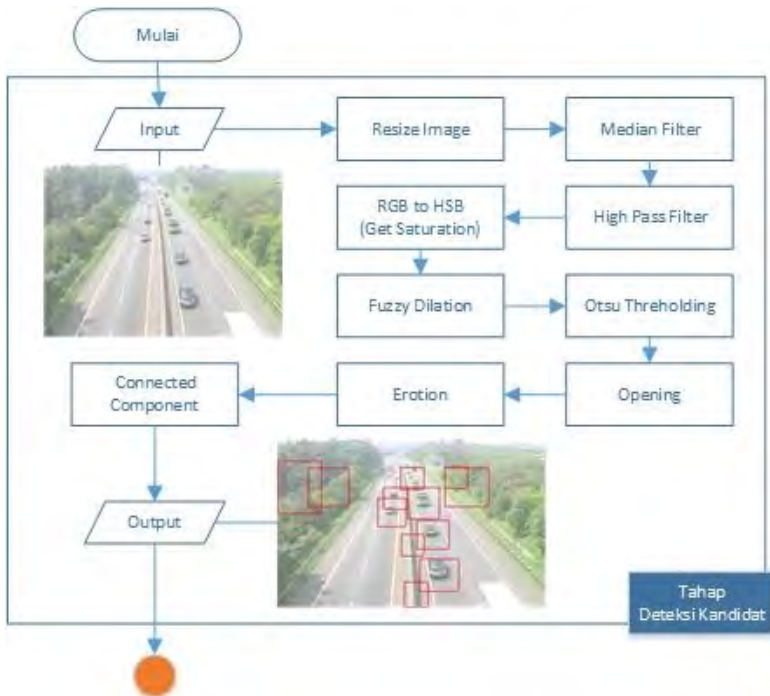
Pada tugas akhir ini akan dibangun sebuah sistem untuk melakukan penghitungan mobil pada sebuah citra *CCTV* menggunakan pustaka Apache Spark, ImgScalr. Proses-proses yang terlibat di dalam implementasi sistem ini meliputi tahap deteksi *area* mobil, tahap ekstraksi fitur, dan tahap *training* sekaligus tahap klasifikasi.

Tahap deteksi kandidat mobil dilakukan pertama kali. Pada tahap ini dilakukan operasi morfologi untuk mencari *area* mobil dengan melacak lampu depan mobil. Tahap ini beberapa objek yang menjadi kandidat mobil. Operasi morfologi yang digunakan yaitu operasi *Fuzzy morphology*, operasi ini digunakan untuk membuat kontras antara mobil yang terang (*foreground*) dengan objek yang tidak terang (*background*). Setelah itu dilakukan metode *Otsu* untuk mendapatkan citra biner. Citra biner yang dihasilkan dari metode *otsu* masih terdapat banyak *noise* yang kemudian dilakukan operasi *opening* dan seleksi luas objek untuk meminimalisir *noise* tersebut.

Tahap ekstraksi fitur merupakan tahap dimana didapatkan fitur-fitur dari tiap objek (data testing) maupun fitur data training, tahapan ini menggunakan Haar wavelet transform dengan tingkat kedalaman 3 level, apabila menggunakan citra RGB untuk di ekstraksi tiap objek didapatkan 54 fitur, apabila menggunakan citra dengan derajat keabuan didapatkan 18 fitur. Setelah didapatkan fitur-fitur tersebut akan dilakukan normalisasi. Hasil dari tahap ini

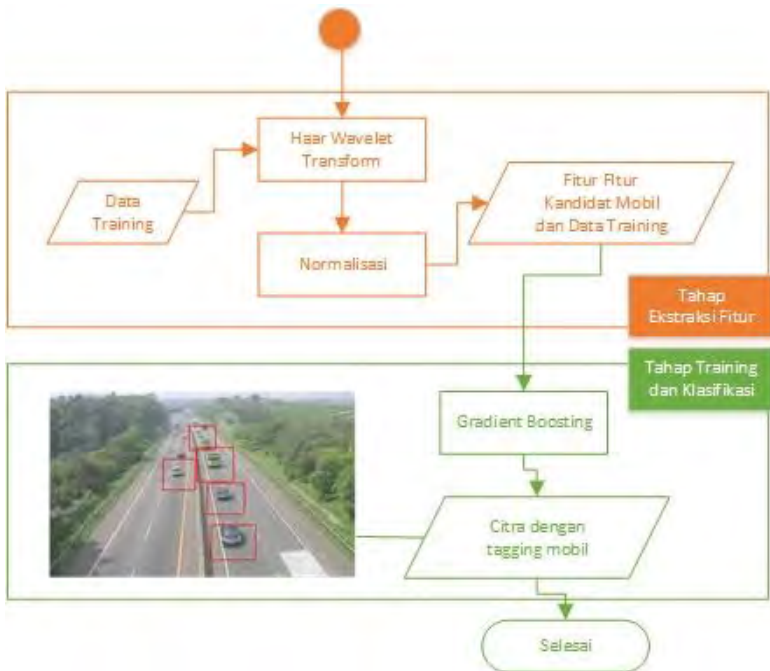
adalah fitur-fitur yang siap digunakan untuk tahap training sekaligus tahap klasifikasi

Tahap *training* sekaligus tahap klasifikasi merupakan tahapan terakhir. Tahap mengolah fitur-fitur dari citra CCTV mobil sebagai sampel positif serta citra yang bukan mobil sebagai sampel negatif untuk kemudian dijadikan sebagai *classifier* dan tahap mengklasifikasikan data *testing*. Tahap ini menggunakan *Gradient Boosting* sebagai algoritma untuk melatih *classifier*. Hasil dari tahap ini adalah decision tree, dimana secara langsung juga dapat mengklasifikasikan data uji. Diagram alir dari keseluruhan proses sistem ditunjukkan oleh Gambar 3.1 dan Gambar 3.2. Gambar Diagram Alir Implementasi Metode Secara Umum 2



**Gambar 3.1 Gambar Diagram Alir Implementasi Metode Secara Umum 1**





**Gambar 3.2 Gambar Diagram Alir Implementasi Metode Secara Umum 2**

### 3.2. Perancangan Data

Pada subbab ini akan dibahas mengenai perancangan data yang merupakan bagian penting karena data sebagai objek yang akan diolah oleh perangkat lunak dalam tugas akhir ini dan menghasilkan sebuah informasi. Data yang akan digunakan pada sistem ini adalah data *training*, data masukan (*input*), data proses, dan data keluaran (*output*) yang memberikan hasil pengolahan sistem ini untuk pengguna.

#### 3.2.1. Data Training

Data *training* yang digunakan berupa sampel positif dan juga sampel negatif. Sampel positif berupa citra mobil dengan

ukuran 100x100 piksel. Sedangkan untuk sampel negatif berupa citra *background* dengan ukuran 100x100, *background* dapat berupa citra yang bukan mobil, bisa seperti jalanan, motor, pohon dsb. Contoh dapat dilihat pada Gambar 3.3

Pembuatan sampel ini dilakukan secara manual dengan cara memotong gambar *CCTV* yang terdapat bagian mobil. Data *training* ini yang kemudian digunakan dalam tahap ekstraksi fitur, training dan klasifikasi.

Pada Gambar 3.3 ditampilkan 36 dari 538 sampel positif, serta 36 dari 1100 sampel negatif dari keseluruhan data *training* yang digunakan pada sistem penghitungan mobil ini.



**Gambar 3.3 Contoh Sampel Negatif (i) dan Positif (ii)**

### 3.2.2. Data Masukan

Data masukan berupa citra *CCTV* yang didapatkan melalui screen capture live streaming di situs jasa marga ([http://www.jasamargalive.com/jsmm\\_/index.php](http://www.jasamargalive.com/jsmm_/index.php)) dengan ukuran 300 x 300 piksel dan memiliki *image channel* bisa berupa RGB maupun dalam *grayscale*. Contoh citra masukan yang digunakan sebagai data masukan ditunjukkan oleh Gambar 3.4

### 3.2.3. Data Proses

Data proses adalah data yang digunakan selama proses berjalannya sistem yang merupakan hasil pengolahan dari data masukan untuk diproses kembali menjadi data keluaran di tahap

selanjutnya. Data proses yang digunakan dalam proses ini ditunjukkan oleh Tabel 3.1



**Gambar 3.4 Citra CCTV sebagai Data Masukan**

### **3.2.4. Data Keluaran**

Data keluaran dari sistem ini berupa citra masukan dengan tambahan informasi berupa *tag* mobil. Data keluaran dari hasil implementasi sistem ini ditunjukkan pada Gambar 3.5.



**Gambar 3.5 Hasil Keluaran**

**Tabel 3.1 Data Proses**

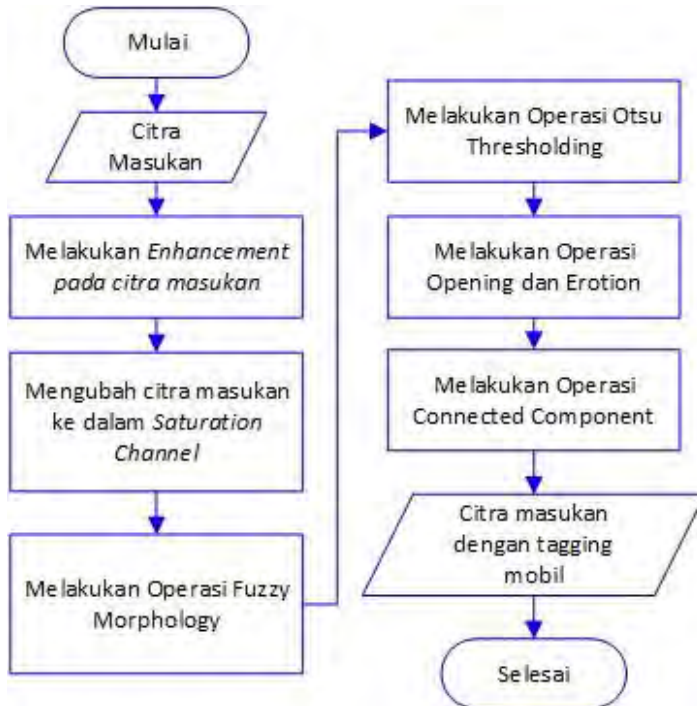
No.	Nama Data	Keterangan
1	Image	Data masukan
2	Method.SPEED	Merupakan parameter <i>imgscalr</i> untuk mengubah ukuran secepat mungkin
3	OP_ANTIALIAS	Merupakan parameter <i>imgscalr</i> untuk <i>mengenhance</i> gambar
4	STRUCTURING_ELEMENT_SHAPE	Parameter class structuring element untuk menentukan bentuk structuring element
5	width	Merupakan atribut dari kelas DetailObj sebagai nilai lebar dari objek.
6	height	Merupakan atribut dari kelas DetailObj sebagai nilai tinggi dari objek.
7	trainingpath	Variabel yang menyimpan lokasi file data training
8	path	Variabel yang menyimpan lokasi file data testing
9	BoostingStrategy	<i>Class Gradient Boosting</i>
10	setNumIterations	Menentukan iterasi dari <i>gradient boosting</i>
11	setNumClasses	Menentukan jumlah <i>class output</i>
12	setMaxDepth	Menentukan tingkat kedalaman <i>tree</i>
13	train	Fungsi untuk melatih data training
14	mapToPair	Menentukan <i>class</i> dari data testing

### 3.3. Perancangan Proses

Pada subbab ini akan dibahas mengenai perancangan proses yang dilakukan untuk memberikan gambaran secara rinci pada setiap alur implementasi metode pada sistem penghitungan mobil. Alur tersebut nantinya akan digunakan dalam tahap implementasi sistem penghitungan mobil.

### 3.3.1. Tahap Deteksi Kandidat Mobil

Tahap deteksi kandidat mobil terdapat beberapa proses pada tahap ini, yaitu mengubah citra masukan menjadi *grayscale*, *Mengenhance* citra masukan, penerapan *fuzzy Morphology*, penerapan *Otsu Thresholding* dan penerapan *Opening*. Diagram alir mengenai tahap ini dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Alir Tahap Deteksi Area Mobil

#### 3.3.1.1. *Enhancement* pada Citra Masukan

*Enhancement* pada citra masukan digunakan untuk meningkatkan kualitas citra masukan. Operasi median filter pada citra masukan merupakan proses pertama pada tahap *enhancement*. Median filter digunakan untuk menghilangkan *salt and pepper noise*. Hasil dari median filter dapat dilihat

pada Gambar 3.7. Kemudian dilakukan operasi high pass filter supaya objek lebih mudah dikenali. Hasil dari high pass filter dapat dilihat pada Gambar 3.8. Setelah itu output dari proses ini digunakan pada proses selanjutnya.



**Gambar 3.7 Citra Hasil Operasi *Median Filter***



**Gambar 3.8 Citra Hasil Operasi *High Pass Filter***

#### **3.3.1.2. Pengubahan Citra ke *Saturation Channel***

Pengubahan citra masukan ke *saturation channel* merupakan proses setelah *enhancement*. Pemilihan *saturation channel* ini karena memiliki hasil yang paling bagus terhadap channel channel lain dan hanya memiliki 1 lapis. Selain itu, alasan penggunaan citra karena pada operasi morfologi

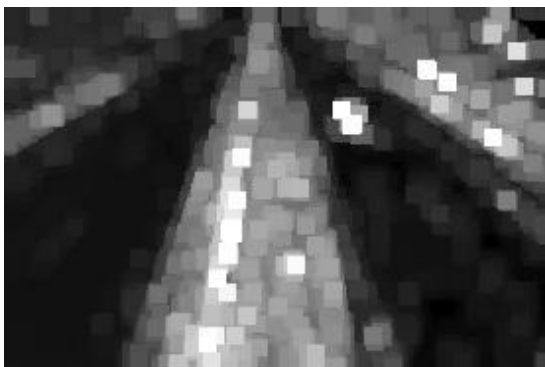
selanjutnya tidak membutuhkan banyak channel. Hasil *saturation channel* dapat dilihat pada Gambar 3.9.



**Gambar 3.9 Citra Hasil *Saturation Channel***

#### **3.3.1.3. *Fuzzy Morphology***

Fuzzy Morphology merupakan tahapan penting pada tugas akhir ini. Operasi yang dilakukan pada tahap ini adalah Fuzzy Dilation. Pada dasarnya dilation digunakan dalam derajat citra biner, dilation tidak berlaku apabila terjadi pada derajat keabuan (bukan biner), maka dari itu digunakannya fuzzy, digunakan untuk menetapkan warna yang digunakan dalam operasi dilation. Hasil *fuzzy morphology* dapat dilihat pada Gambar 3.10.



**Gambar 3.10 Citra Hasil *Fuzzy Dilation***

#### 3.3.1.4. Penerapan Threshold

Operasi *Thresholding* citra merupakan proses mengubah citra *grayscale* menjadi citra biner atau hitam putih. Proses ini akan diketahui daerah yang akan menjadi *background* maupun yang menjadi objek. Hasil dari operasi ini dapat dilihat pada Gambar 3.11.



Gambar 3.11 Citra Hasil Operasi *Otsu Thresholding*

#### 3.3.1.5. Penerapan Operasi *Opening dan Erosion*

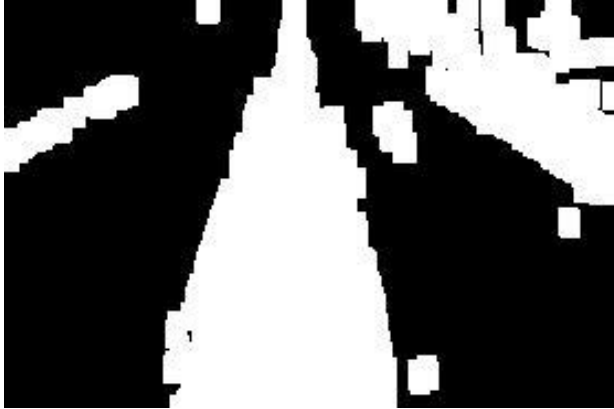
Setelah mendapatkan citra biner dari operasi *thresholding*, maka dapat dilihat jika banyak objek *noise* yang muncul. Objek *noise* ini dapat diminimalisir dengan melakukan operasi *opening* dan *erosion*. Celah (*gap*) yang terdapat setelah hasil *thresholding* akan hilang sesuai dengan ukuran kernel (*structuring element*). Semakin besar ukuran kernel, maka semua objek akan dikikis dengan objek yang lebih besar. Jika objek lebih kecil dari ukuran kernel, maka objek tersebut akan hilang. Contoh dari operasi ini dapat dilihat pada Gambar 3.12.

#### 3.3.1.6. *Connected Component*

*Connected Component* merupakan tahap akhir dalam mendapatkan kandidat mobil, hasil dari operasi *opening*



didapatkan kandidat mobil, dari kandidat kandidat tersebut didapatkan koordinat koordinat tiap kandidat dalam suatu variabel.



**Gambar 3.12 Hasil dari Operasi *Opening Dan Erosion***

### **3.3.2. Tahap Ekstraksi Fitur**

Pada tahap ini dilakukan ekstraksi fitur, baik data testing maupun data training, ekstraksi fitur ini menggunakan *haar wavelet transform* dengan tingkat kedalaman 3 level.

#### **3.3.2.1. Ekstraksi Fitur Menggunakan Haar Wavelet Transform.**

Pada tahap ini, citra data training akan di ekstraksi fitur satu persatu, begitu juga data testing yang didapat dari proses sebelumnya. Untuk mengekstraksi ciri tekstur, dilakukan proses averaging untuk mendapatkan bagian dengan frekuensi rendah dan dilakukan proses differencing untuk mendapatkan bagian dengan frekuensi tinggi. Rumus untuk mencari nilai everaging ( $a_i$ ) dan differencing ( $c_i$ ) dapat dilihat pada persamaan 3.1 dan 3.2.

$$a_i = \frac{S_i + S_{i+1}}{2} \quad (3.1)$$

$$c_i = \frac{S_i - S_{i+1}}{2} \quad (3.2)$$

Berikut langkah-langkah melakukan tranformasi citra dengan haar wavelet transform :

1. Dilakukan tranformasi pertama dengan membagi 2 matriks citra berdasarkan kolom, sehingga menghasilkan 2 *sub-band*, sebelah kiri frekuensi rendah dan sebelah kanan dengan frekuensi tinggi.
2. Selanjutnya matriks citra dibagi 2 berdasarkan baris dan menghasilkan 4 *sub-band*.
3. Proses transoformasi kembali ke langkah 1 dan 2 dilakukan pada sub-band dengan frekuensi rendah secara terus menerus bergantung pada tingkat kedalaman yang digunakan.

Setelah dilakukan tranformasi citra, tiap tiap sub-band akan di ekstraksi fiturnya menggunakan rumus standar deviasi dan nilai rata-rata. Rumus standar deviasi ( $s$ ) dan nilai rata-rata ( $\bar{X}$ ) dapat dilihat pada persamaan 3.3 dan 3.4

$$s = \sqrt{\frac{\sum X_i^2 - \frac{(\sum X_i)^2}{n}}{n - 1}} \quad (3.3)$$

$$\bar{X} = \frac{\sum X_i}{n} \quad (3.4)$$

### 3.3.3. Tahap Training Data dan Klasifikasi

Pada tahap ini merupakan tahap terakhir, hasil dari ekstraksi fitur akan latih dan diklasifikasikan dalam 2 class, mobil atau bukan mobil. Tahap ini menggunakan Gradient Boosting.

#### 3.3.3.1. Training Data dan Klasifikasi menggunakan Gradient Boosting.

Hasil dari ekstraksi fitur akan menjadi masukan dalam tahap training data dan klasifikasi. Fitur fitur *data training* akan di

latih terlebih dahulu menggunakan *Gradient Boosting*. Gradient boosting secara keseluruhan yaitu melatih pohon keputusan dengan mengurangi *loss function* pada *data training*. Dalam setiap iterasi, bertujuan untuk memprediksi label dari setiap training kemudian membandingkan prediksinya dengan label sebenarnya. Dataset tersebut di beri label untuk memberikan bobot pada data dengan prediksi yang buruk. Mekanisme dalam kasus pemberian label didefinisikan oleh *loss function*. Algoritma dari *Gradient Boosting* yang diterapkan dapat dilihat seperti berikut :

1. Inisiasi model dengan nilai konstan menggunakan persamaan 3.5

$$F_0(x) = \operatorname{argmin} \sum_{i=1}^n L(y_i, r) \quad (3.5)$$

2. For m=1 to M :

- a. Hitung pseudo residuals menggunakan persamaan 3.6

$$r_m = - \left[ \frac{\partial L(y, F(x))}{\partial F(x)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1..n \quad (3.6)$$

- b. Sesuaikan *base-learner*  $h_m(x)$  dengan pseudo-residuals.

- c. Hitung perkalian  $r_m$  dengan menyelesaikan masalah optimasi menggunakan persamaan 3.7

$$\gamma_m = \operatorname{argmin} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \quad (3.7)$$

- d. Perbarui model menggunakan persamaan 3.8

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x) \quad (3.8)$$

3. *Output*  $F_M(x)$

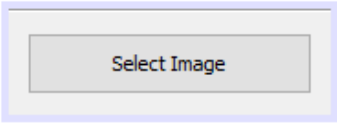
Setelah training dilakukan dan menghasilkan model decision tree, setiap data *testing*, satu persatu akan diklasifikasikan

berdasarkan *decision tree* tersebut. Sehingga didapatkan objek mobil dan bukan objek mobil.

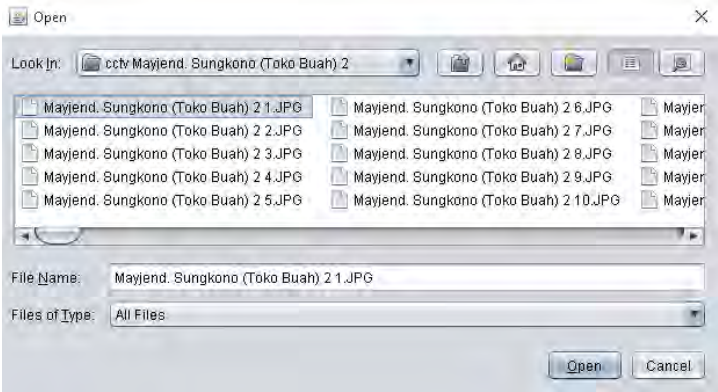
**3.4. Perancangan Antarmuka Perangkat Lunak**

Pada subbab ini akan dibahas mengenai perancangan antarmuka perangkat lunak yang bertujuan untuk dapat mempermudah interaksi antara perangkat lunak dengan pengguna. Sistem ini memiliki 3 halaman.

Halaman pertama merupakan halaman untuk memilih gambar. Pada halaman ini pengguna memilih gambar yang akan diproses, dapat dilihat pada Gambar 3.13. Setelah tombol ditekan akan menampilkan *widget file chooser* yang ada pada Gambar 3.14. Penjelasan mengenai atribut-atribut yang terdapat pada halaman pertama dapat dilihat pada Tabel 3.2



**Gambar 3.13 Rancangan Halaman Pertama**

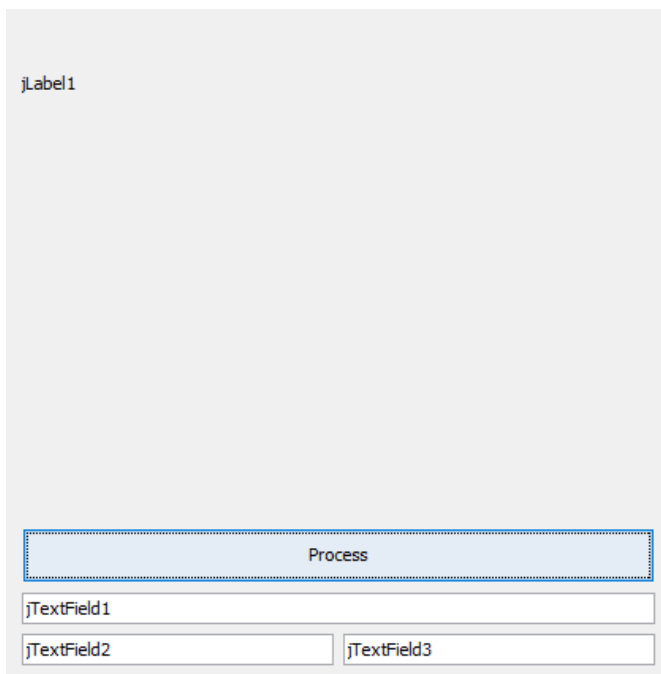


**Gambar 3.14 Widget File Chooser**

**Tabel 3.2 Spesifikasi atribut antarmuka halaman pertama**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	jButton1	Button	Pilihan aksi untuk menampilkan widget file chooser	Button action

Halaman kedua merupakan halaman untuk menampilkan gambar yang dipilih beserta informasinya dan melanjutkan pada proses selanjutnya. Halaman ini dapat dilihat pada Gambar 3.15. Setelah tombol ditekan akan lanjut pada proses selanjutnya. Penjelasan mengenai atribut-atribut yang terdapat pada halaman pertama dapat dilihat pada Tabel 3.3



**Gambar 3.15 Rancangan Halaman Kedua**

**Tabel 3.3 Spesifikasi Atribut antarmuka Halaman Kedua**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	jButton1	<i>Button</i>	Pilihan aksi untuk memproses gambar	<i>Button action</i>
2	(jTextField1)	<i>Text area</i>	Menampilkan lokasi gambar masukan	Lokasi gambar
3	(jTextField2)	<i>Text area</i>	Menampilkan ukuran lebar gambar	Lebar
4	(jTextField3)	<i>Text area</i>	Menampilkan ukuran tinggi gambar	Tinggi
5	jLabel1	<i>Label</i>	Menampilkan gambar masukan yang orisinal	<i>Image</i>

Halaman ketiga merupakan halaman untuk menampilkan gambar masukan, gambar hasil proses *fuzzy morphology* dan gambar hasil proses *gradient boosting*. Halaman ini dapat dilihat pada Gambar 3.16. Penjelasan mengenai atribut-atribut yang terdapat pada halaman pertama dapat dilihat pada Tabel 3.4

**Gambar 3.16 Rancangan Halaman Ketiga**

**Tabel 3.4 Spesifikasi Atribut antarmuka Halaman Ketiga**

<b>No</b>	<b>Nama Atribut Antarmuka</b>	<b>Jenis Atribut</b>	<b>Kegunaan</b>	<b>Jenis Masukan / Keluaran</b>
<b>1</b>	jLabel1	<i>Label</i>	Menampilkan gambar setelah gradient boosting	<i>Image</i>
<b>2</b>	jLabel2	<i>Label</i>	Menampilkan gambar masukan.	<i>Image</i>
<b>3</b>	jLabel3	<i>Label</i>	Menampilkan gambar setelah fuzzy morphology	<i>Image</i>
<b>4</b>	jLabel4	<i>Label</i>	Menampilkan teks Input Gambar	<i>Text</i>
<b>5</b>	jLabel5	<i>Label</i>	Menampilkan teks Hasil Fuzzy Morphology	<i>Text</i>
<b>6</b>	jLabel6	<i>Label</i>	Menampilkan teks Hasil Gradient Boosting	<i>Text</i>

*[Halaman ini sengaja dikosongkan]*



## BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah bahasa pemrograman Java dengan pustaka Apache Spark dan ImgScalr.

### 4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti ditampilkan pada Tabel 4.1.

**Tabel 4.1 Lingkungan Implementasi Sistem**

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i5-2450M CPU @ 2.50GHz (4 CPUs) , ~2.4GHz Memori: 8192MB
Perangkat lunak	Sistem Operasi: Microsoft Windows 10 64-bit Perangkat Pengembang: Netbeans 8.0.2 Perangkat Pembantu: Notepad, Sublime, Microsoft Excel 2013, Microsoft Word 2013

### 4.2. Implementasi Proses

Implementasi proses dilakukan berdasarkan perancangan proses yang sudah dijelaskan pada bab analisis dan perancangan.

#### 4.2.1. Implementasi Tahap Deteksi Kandidat Mobil

Subbab ini membahas mengenai implementasi tahap deteksi kandidat mobil yang akan digunakan pada tahap selanjutnya. Implementasi tahap ini dilakukan dengan *mengenhance* gambar masukan dengan *median filter* dan *high pass*

*filter* kemudian dilakukan proses Fuzzy Morphology yaitu menggunakan *channel saturation* lalu operasi *fuzzy dilation*, *otsu thresholding*, *opening* dan *erotion*, setelah *fuzzy morphology* dilakukan *connected component* untuk mendapatkan kandidat mobil.

Proses *enhancement* dilakukan sesuai dengan input yang ada, operasi yang digunakan tergantung pada input yang diberikan. Untuk tugas akhir kali ini, digunakanlah median filter dan high pass filter, *resize image*, dan penambahan *brightness*. *Resize image* digunakan apabila hasil tanpa metode ini mendapatkan hasil yang cukup rendah sehingga diperlukan *resize image* supaya didapatkan mobil yang memiliki ukuran cukup kecil. Kode Sumber 4.1 Kode Program *Median Filter* dan Kode Sumber 4.2 untuk *high pass filter channel* merah. Parameter yang diperlukan pada median filter adalah matriks nilai intensitas, bentuk *structure element* yang digunakan, ukuran *structure element*, dan R/G/B. untuk parameter *high pass filter*, hanya memerlukan parameter matriks nilai intensitas dan bentuk *structure element*.

1	public int median(Color[][] val, int sSize, STRUCTURING_ELEMENT_SHAPE shape, String RGB)
2	int median = 1, count = 0;
3	int[] R = new int[sSize];
4	if(shape == STRUCTURING_ELEMENT_SHAPE.SQUARE    shape == STRUCTURING_ELEMENT_SHAPE.ROUND)
5	sSize = sSize * sSize; median = 4;
6	for(int i = 0; i < val.length ; i++)
7	for(int j = 0 ; j < val[i].length ; j++)
8	if(RGB == "R"){
9	if(val[i][j] == null)
10	continue
11	R[count] = val[i][j].getRed();
12	count++;
13	Arrays.sort(R);
14	median = R[median];
15	return median;

**Kode Sumber 4.1 Kode Program Median Filter Channel Merah**

1	public double[] HPF(Color[][] val, STRUCTURING_ELEMENT_SHAPE shape)
2	double[] hpf = new double[3];
3	hpf[0] = 0;
4	for(int x = 0 ; x < 3 ; x++)
5	for(int y = 0 ; y < 3 ; y++)
6	hpf[0] = hpf[0] + (val[x][y].getRed() * matrix[x][y]);
7	if (hpf[0] > 255)
8	hpf[0] = 255;
9	else if (hpf[0] < 0)
10	hpf[0] = 0;
11	return hpf;

**Kode Sumber 4.2 Kode Program High Pass Filter Channel Merah**

Setelah dilakukan *enhancement*, gambar akan diolah dengan *fuzzy morphology*, *channel* yang digunakan adalah *channel saturation*. Kode sumber untuk mendapatkan *channel saturation* dapat dilihat pada Kode Sumber 4.3. Kemudian dilakukan *fuzzy dilation*, *fuzzy* digunakan untuk menentukan nilai derajat keabuan yang ditetapkan untuk dilakukannya *dilation*, sehingga bisa digunakan citra dengan derajat keabuan. Kode sumber utama *fuzzy dilation* dapat dilihat pada Kode Sumber 4.4. *Fuzzy* untuk menentukan tingkat keabuan yang akan digunakan dalam *dilation* dapat dilihat pada Kode Sumber 4.5

1	public BufferedImage Saturation(BufferedImage img)
2	int width = img.getWidth();
3	int height = img.getHeight();
4	BufferedImage HSBImage = new BufferedImage(width, height, img.getType());
5	for (int i=0;i<height;i++){
6	for(int j=0;j<width;j++){
7	float[] hsb = Color.RGBtoHSB(Red, Green, Blue, null);
8	float saturation = hsb[1];
9	Color newColor = new Color(saturation,saturation,saturation);
10	HSBImage.setRGB(j, i, newColor.getRGB());
11	return HSBImage;

**Kode Sumber 4.3 Kode Program untuk Mendapatkan Channel Saturation**

1	public int DKD(Color[][] val,int sSize, STRUCTURING_ELEMENT_SHAPE shape, int[][] matrix)
2	if(shape == STRUCTURING_ELEMENT_SHAPE.SQUARE    shape == STRUCTURING_ELEMENT_SHAPE.ROUND)
3	sSize = sSize * sSize;
4	int[] temp = new int[sSize];
5	int count = 0;
6	for(int i = 0; i < val.length ; i++)
7	for(int j = 0 ; j < val[i].length ; j++)
8	if(val[i][j] == null) continue;
9	temp[count] = Conjunction(matrix[i][j],val[i][j].getRed());
10	count++;
11	int max = temp[0], end = temp.length, v = 0;
12	for (int i = 1; i < end; i++)
13	if(max < temp[i]) max = temp[i];
14	return max;

**Kode Sumber 4.4 Kode Program Fungsi Utama Fuzzy Dilation**

1	public int Conjunction(int a, int val)
2	int hasil;
3	int b = 1-a;
4	if ((double)val/255 <= 1-a)
5	hasil = 0;
6	else
7	hasil = val;
8	return hasil;

**Kode Sumber 4.5 Kode Program Fuzzy untuk Menentukan Tingkat Keabuan**

Setelah dilakukan *fuzzy dilation*, operasi yang dilakukan adalah *otsu thresholding*, parameter yang dibutuhkan adalah citra gambar dengan derajat keabuan dan akan me-*return* citra gambar dengan derajat biner. Kode sumber otsu untuk mendapatkan nilai threshold secara otomatis dapat dilihat pada Kode Sumber 4.6 Kode Program *Otsu*

1	public int OtsuThreshold(BufferedImage img)
2	int[] histogram = Histogram(img)
3	int total = img.getWidth() * img.getHeight()
4	sum = 0; sumB = 0;
5	for (int i = 0; i < 256; i++)
6	sum += i * histogram[i]
7	wB = 0; //weight background
8	wF = 0; //weight foreground
9	varMax = 0;
10	int threshold = 0;
11	for (int i = 0; i < 256; i++)
12	wB += histogram[i];
13	if (wB == 0) continue;
14	wF = total - wB;
15	if (wF == 0) break;
16	sumB += (float) (i * histogram[i]);
17	float mB = sumB / wB; //mean Background
18	float mF = (sum - sumB) / wF; //mean foreground
19	float varBetween =
	(float) wB * (float) wF * (mB - mF) * (mB - mF);
20	if (varBetween > varMax)
21	varMax = varBetween;
22	threshold = i;
23	return threshold

**Kode Sumber 4.6 Kode Program *Otsu* untuk Mendapatkan Nilai *Threshold***

Selanjutnya adalah operasi opening dan erotion, parameter yang digunakan hampir sama dengan median filter dan high pass filter yaitu, citra dengan derajat keabuan, ukuran dan bentuk structure element dilation dan erotion, banyaknya dilakukan erotion dan dilation. Kode sumber opening dapat dilihat pada Kode Sumber 4.7, Kode sumber untuk mendapatkan nilai intensitas erotion dapat dilihat pada Kode Sumber 4.8 , Kode sumber untuk mendapatkan nilai intensitas dilation dapat dilihat pada Kode Sumber 4.8,

1	public BufferedImage Opening(BufferedImage img,int shapeSize, StructuringElement.STRUCTURING_ELEMENT_SHAPE shape, StructuringElement.STRUCTURING_ELEMENT_SHAPE shape2, int erosion, int dilation)
2	for(int a=0; a<erosion;a++)
3	img = new Erosion().Erosion(img,shapeSize,shape);
4	for(int a=0; a<dilation;a++)
5	img = new Dilation().Dilation(img,shapeSize,shape2);
6	return img;

**Kode Sumber 4.7 Kode Program *Opening***

1	public int min(Color[][] val,int sSize,StructuringElement.STRUCTURING_ELEMENT_SHAPE shape)
2	if(shape == STRUCTURING_ELEMENT_SHAPE.SQUARE    shape == STRUCTURING_ELEMENT_SHAPE.ROUND)
3	sSize = sSize * sSize;
4	int[] R = new int[sSize];
5	int count = 0;
6	for(int i = 0; i < val.length ; i++){
7	for(int j = 0 ; j < val[i].length ; j++){
8	if(val[i][j] == null) continue;
9	R[count] = val[i][j].getRed();
10	count++;
11	int min = 256;
12	int end = val.length;
13	int v = 0;
14	for (int i = 0; i < end; i++)
15	if(R[i] < 0)
16	v = 256+R[i];
17	else
18	v = R[i];
19	if (v < min) min = v;
20	return min;

**Kode Sumber 4.8 Kode untuk Mendapatkan Nilai Intensitas *Eroton***

1	public int max(Color[][] val,int sSize,StructuringElement.STRUCTURING_ELEMENT_SHAPE shape)
2	if(shape == STRUCTURING_ELEMENT_SHAPE.SQUARE    shape == STRUCTURING_ELEMENT_SHAPE.ROUND)
3	sSize = sSize * sSize;
4	int[] R = new int[sSize];
5	int count = 0;
6	for(int i = 0; i < val.length ; i++){
7	for(int j = 0 ; j < val[i].length ; j++){
8	if(val[i][j] == null) continue;
9	R[count] = val[i][j].getRed();
10	count++;
11	int max = R[0];
12	int end = R.length;
13	int v = 0;
14	for (int i = 1; i < end; i++) {
15	if (R[i] < 0)
16	v = 256 + R[i];
17	else
18	v = R[i];
19	if (v > max) max = v;
20	return max;

**Kode Sumber 4.9 Kode untuk Mendapatkan Nilai Intensitas  
*Dilation***

Setelah itu dilakukan *connected component*, untuk menggunakan connected component dapat mengimplementasikan Kode Sumber 4.10.

1	aImage = new ConnectedComponent().ConnectedComponent(image, originalimage);
---	--

**Kode Sumber 4.10 Fungsi *Connected Component***

#### 4.2.2. Implementasi Tahap Ekstraksi Fitur

Pada tahap ini akan dilakukan ekstraksi fitur, baik fitur *data training* maupun *data testing*, ekstraksi fitur menggunakan haar wavelet transform dengan tingkat kedalaman 3 level.

Implementasi haar wavelet transform dilakukan pada channel RGB, sehingga tiap channel melakukan transformasi citra. Kode program implementasi *haar wavelet transform channel* merah dapat dilihat pada Kode Sumber 4.11.

1	for (int i = 0; i < cycles; i++)
2	w /= 2;
3	for (int j = 0; j < h; j++)
4	for (int k = 0; k < w; k++)
5	double aR = dsR[j][2 * k];
6	double bR = dsR[j][2 * k + 1];
7	double addR = aR + bR;
8	double subR = aR - bR;
9	double avgAddR = addR / 2;
10	double avgSubR = subR / 2;
11	tempdsR[j][k] = avgAddR;
12	tempdsR[j][k + w] = avgSubR;
13	for (int j = 0; j < h; j++)
14	for (int k = 0; k < w; k++)
15	dsR[j][k] = tempdsR[j][k];
16	dsR[j][k + w] = tempdsR[j][k + w];
17	h /= 2;
18	for (int j = 0; j < w; j++)
19	for (int k = 0; k < h; k++)
20	double aR = dsR[2 * k][j];
21	double bR = dsR[2 * k + 1][j];
22	double addR = aR + bR;
23	double subR = aR - bR;
24	double avgAddR = addR / 2;
25	double avgSubR = subR / 2;
26	tempdsR[k][j] = avgAddR;
27	tempdsR[k + h][j] = avgSubR;
28	for (int j = 0; j < w; j++)
29	for (int k = 0; k < h; k++)
30	dsR[k][j] = tempdsR[k][j];
31	dsR[k + h][j] = tempdsR[k + h][j];

**Kode Sumber 4.11 Kode Program Haar Wavelet Transform Channel Merah**



Setelah dilakukan transformasi citra maka dilakukan ekstraksi fitur. Kode program ekstraksi fitur pada *channel* merah dapat dilihat pada Kode Sumber 4.12.

1	for (int a = 0; a < cycles; a++)
2	int index = 5 + a * 18;
3	double yR = 0;
4	double y2R = 0;
5	for (int i = 0; i < height/2; i++)
6	for (int j = width/2; j < width; j++)
7	yR = yR + dsR[i][j];
8	y2R = y2R + (dsR[i][j] * dsR[i][j]);
9	double pembilang = (y2R) - ((yR*yR)/((height*width)/4));
10	double stDeviasi = sqrt(pembilang / (((height*width)/4) - 1));
11	double mean = yR / ((height*width)/4);
12	Fitur.get(Fitur.size()-1).add(index + 0,stDeviasi);
13	Fitur.get(Fitur.size()-1).add(index + 1,mean);
14	yR = 0;
15	y2R = 0;
16	for (int i = height/2; i < height; i++)
17	for (int j = 0; j < width/2; j++)
18	yR = yR + dsR[i][j];
19	y2R = y2R + (dsR[i][j] * dsR[i][j]);
20	pembilang = (y2R) - ((yR*yR)/((height*width)/4));
21	stDeviasi = sqrt(pembilang / (((height*width)/4) - 1));
22	mean = yR / ((height*width)/4);
23	Fitur.get(Fitur.size()-1).add(index + 6,stDeviasi);
24	Fitur.get(Fitur.size()-1).add(index + 7,mean);
25	yR = 0;
26	y2R = 0;
27	for (int i = height/2; i < height; i++)
28	for (int j = width/2; j < width; j++)
29	yR = yR + dsR[i][j];
30	y2R = y2R + (dsR[i][j] * dsR[i][j]);
31	pembilang = (y2R) - ((yR*yR)/((height*width)/4));
32	stDeviasi = sqrt(pembilang / (((height*width)/4) - 1));
33	mean = yR / ((height*width)/4);
34	Fitur.get(Fitur.size()-1).add(index + 12,stDeviasi);
35	Fitur.get(Fitur.size()-1).add(index + 13,mean);

**Kode Sumber 4.12 Kode Program Ekstraksi Fitur pada *Channel* Merah**

Setelah didapatkan fitur dari tiap tiap channel baik data testing maupun data training, dilakukan normalisasi. Kode program implementasi normalisasi dapat dilihat pada Kode Sumber 4.13.

1	for (int x = 1; x < listFitur.get(0).size();)
2	Double min = 0.0;
3	Double max = 0.0;
4	for (int y = 0; y < listFitur.size();y++ )
5	if(y == 0)
6	for (int a = 0; a < fitur.size() ; a++)
7	if(a== 0) min = fitur.get(a).get(x+4);
8	else
9	if(min > fitur.get(a).get(x+4))
10	min = fitur.get(a).get(x+4);
11	if(min > listFitur.get(y).get(x))
12	min = listFitur.get(y).get(x);
13	for (int y = 0; y < listFitur.size();y++ )
14	if(y == 0)
15	for (int a = 0; a < fitur.size() ; a++)
16	Double value = fitur.get(a).get(x+4) - min;
17	fitur.get(a).set(x+4,value);
18	Double value = listFitur.get(y).get(x) - min;
19	listFitur.get(y).set(x,value);
20	for (int y = 0; y < listFitur.size();y++ )
21	if(y == 0)
22	for (int a = 0; a < fitur.size() ; a++)
23	if(a== 0) max = fitur.get(a).get(x+4);
24	else
25	if(max < fitur.get(a).get(x+4))
26	max = fitur.get(a).get(x+4);
27	if(max < listFitur.get(y).get(x))
28	max = listFitur.get(y).get(x);
29	for (int y = 0; y < listFitur.size();y++ )
30	if(y == 0)
31	for (int a = 0; a < fitur.size() ; a++)
32	Double value = fitur.get(a).get(x+4) / max;
33	fitur.get(a).set(x+4,value);
34	Double value = listFitur.get(y).get(x) / max;
35	listFitur.get(y).set(x,value);
36	x++;

**Kode Sumber 4.13 Kode Program Normalisasi Fitur**

Keluaran normalisasi fitur akan ditulis pada sebuah file dimana file tersebut nanti dibaca oleh *library gradient boosting* untuk tahap *training data* dan klasifikasi.

#### 4.2.3. Implementasi Training Data dan Klasifikasi

Pada tahap ini akan dilakukan training data dan klasifikasi dengan *gradient boosting* untuk memverifikasi kandidat object mobil menggunakan pustaka Apache Spark [9]. Implementasi proses ini pada sistem dapat dilihat pada Kode Sumber 4.14 dan Kode Sumber 4.15

1	System.setProperty("hadoop.home.dir", "E:\\Teknik Informatika 2012\\Tugas Akhir\\Spark\\hadoop-common-2.2.0-bin-master");
2	SparkConf sparkConf = new SparkConf().setAppName("JavaGradientBoostedTreesClassificationExample").setMaster("local");;
3	JavaSparkContext jsc = new JavaSparkContext(sparkConf);
4	// Load and parse the data file.
5	String datapath = trainingpath;
6	JavaRDD<LabeledPoint> trainingData = MLUtils.loadLibSVMFile(jsc.sc(), datapath).toJavaRDD();
7	//datatesting pasti urut
8	String testingDatapath = path;
9	JavaRDD<LabeledPoint> testData = MLUtils.loadLibSVMFile(jsc.sc(), testingDatapath).toJavaRDD();
10	// Train a GradientBoostedTrees model.
11	// The defaultParams for Classification use LogLoss by default.
12	BoostingStrategy boostingStrategy = BoostingStrategy.defaultParams("Classification");
13	boostingStrategy.setNumIterations(10); // Note: Use more iterations in practice.
14	boostingStrategy.getTreeStrategy().setNumClasses(2);
15	boostingStrategy.getTreeStrategy().setMaxDepth(5);
16	// Empty categoricalFeaturesInfo indicates all features are continuous.
17	Map<Integer, Integer> categoricalFeaturesInfo = new HashMap<Integer, Integer>();
18	boostingStrategy.treeStrategy().setCategoricalFeaturesInfo(categoricalFeaturesInfo);
19	final GradientBoostedTreesModel model

**Kode Sumber 4.14 Implementasi Gradient Boosting (I)**

20	= GradientBoostedTrees.train(trainingData, boostingStrategy);
21	// Evaluate model on test instances and compute test error
22	JavaPairRDD<Double, Double> predictionAndLabel
23	= testData.mapToPair(new PairFunction<LabeledPoint, Double, Double>() {
24	@Override
25	public Tuple2<Double, Double> call(LabeledPoint p) {
26	return new Tuple2<Double, Double>(model.predict(p.features()), p.label());
27	}
28	});
29	ArrayList<Double> out = new ArrayList<>();
30	for (int i = 0; i < predictionAndLabel.count(); i++) {
31	out.add(predictionAndLabel.collect().get(i)._1());
32	}

**Kode Sumber 4.15 Implementasi *Gradient Boosting* (II)**

### 4.3. Implementasi Antarmuka Perangkat Lunak

Implementasi tampilan antarmuka pengguna pada perangkat lunak berbasis *desktop* dan berjalan pada sistem operasi Windows 10. Implementasi menggunakan *GUI Editor* bawaan dari IDE Netbeans 8.0.2. Antarmuka aplikasi ini terdapat 3 halaman.

Halaman pertama digunakan untuk memilih gambar, hanya ada tombol pilih gambar pada halaman ini, tombol ini akan menampilkan *widget file chooser* seperti pada Gambar 3.14. Setelah memilih gambar pada *widget file chooser* akan tampil halaman kedua, dimana menampilkan gambar yang dipilih dan terdapat tombol proses. Ketika tombol proses ditekan, maka proses dari pendeteksian mobil akan dimulai, kode program implementasi dapat dilihat pada Kode Sumber 4.16 Perintah pada Tombol Proses.

1	String filepath = jTextField1.getText();
2	TugasAkhir tugasakhir = new TugasAkhir(filepath);
3	this.setVisible(false);

**Kode Sumber 4.16 Perintah pada Tombol Proses**

Setelah tombol proses ditekan dan selesai melakukan pedeteksian mobil, maka akan muncul halaman terakhir, halaman yang menampilkan gambar dengan tag mobil dan gambar hasil fuzzy morphology. Implementasi pada halaman ini dapat dilihat pada Gambar 4.1



**Gambar 4.1 Implementasi Halaman Terakhir**

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi sistem Deteksi Mobil

#### **5.1. Lingkungan Pengujian**

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor	: Prosesor: Intel® Core™ i5-2450M CPU @ 2.50GHz (4 CPUs) ~2.4GHz
RAM	: 8192 MB
Jenis <i>Device</i>	: <i>Notebook</i>
Sistem Operasi	: Microsoft Windows 10

#### **5.2. Data Uji Coba**

Data yang digunakan untuk uji coba implementasi sistem penghitungan mobil ini adalah citra yang diambil dari *CCTV* pada situs jasmarga dengan ukuran maksimal 400 x 400 piksel dengan *channel* RGB. Data *testing* dan data training yang digunakan adalah data *CCTV* Padalarang, Semarang dan Cipularang untuk skenario satu *CCTV* sedangkan untuk scenario banyak *CCTV*, menggunakan *CCTV* berbagai lokasi. Jumlah data testing pada skenario satu *CCTV*, masing masing *CCTV* menggunakan 10 citra sedangkan Banyak *CCTV* menggunakan 47 citra dari berbagai *CCTV*.

#### **5.3. Skenario Uji Coba**

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu:

1. Skenario pengujian 1: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra satu CCTV menggunakan *Fuzzy Morphology*
2. Skenario pengujian 2: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra banyak CCTV menggunakan *Fuzzy Morphology*
3. Skenario pengujian 3: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra satu CCTV menggunakan *Fuzzy Morphology*, *Gradient Boosting* dan ekstraksi fitur pada *channel RGB*.
4. Skenario pengujian 4: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra banyak CCTV menggunakan *Fuzzy Morphology*, *Gradient Boosting* dan ekstraksi fitur pada *channel RGB*
5. Skenario pengujian 5: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra satu CCTV menggunakan *Fuzzy Morphology*, *Gradient Boosting* dan ekstraksi fitur pada *channel Grayscale*
6. Skenario pengujian 6: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra banyak CCTV menggunakan *Fuzzy Morphology*, *Gradient Boosting* dan ekstraksi fitur pada *channel Grayscale*

#### **5.4. Skenario Pengujian 1: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra satu CCTV menggunakan Fuzzy Morphology.**

Pada skenario ini dilakukan uji coba sistem deteksi mobil menggunakan *Fuzzy Morphology*. Uji coba menggunakan 3 CCTV, yaitu CCTV Padalarang, CCTV Semarang dan CCTV Cipularang, tiap tiap CCTV terdapat 10 citra, ekstraksi fitur dilakukan pada channel RGB, dataset yang digunakan berdasarkan CCTV yang digunakan. dan dilakukan perhitungan nilai *recall* dan nilai *precision* dihitung berdasarkan ketentuan berikut:

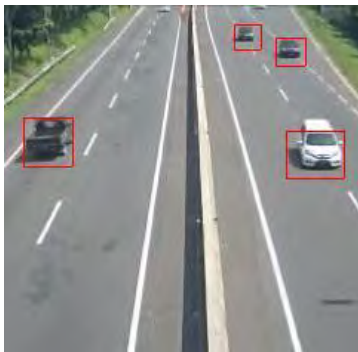
1. *True positif*: sebuah *tag* dengan objek mobil.



2. *False negatif*: sebuah mobil namun tanpa *tag*.
3. *False positif*: sebuah *tag* namun tanpa objek mobil.

Pada CCTV Semarang, skenario ini menggunakan median filter dengan ukuran *structuring element* yaitu 1, high pass filter, Fuzzy dilation dengan *structuring element* yaitu 4, *otsu threshold*, *structuring element* pada operasi *opening* dan erosi yaitu 2. Pada CCTV Cipularang, skenario ini menggunakan citra masukan yang nilai intensitasnya telah dinaikkan sebanyak 50 tiap pixel, nilai *threshold* yaitu 1, Pada CCTV Padalarang, skenario ini menggunakan citra masukan yang lebar ukurannya diperbesar menjadi 600 pixel dan tinggi ukuran mengikuti serta *otsu threshold*.

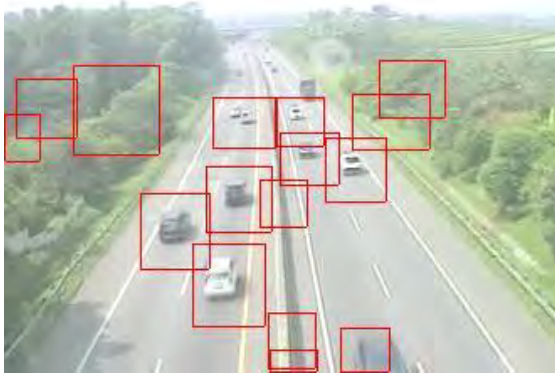
Hasil uji coba ini dapat dilihat pada lampiran. Pada CCTV Semarang, nilai rata-rata *recall* dari *layer* ini adalah 87.9% serta untuk nilai rata-rata *precision* adalah 90.5%. Terdapat total 35 mobil terdeteksi dengan benar, namun tidak mendeteksi 6 mobil. Selain itu ada 4 *tag* dengan yang bukan mobil. Contoh hasil CCTV Semarang dilihat pada Gambar 5.1 .



**Gambar 5.1 Hasil Pengujian pada Citra CCTV Semarang Menggunakan Fuzzy Morphology**

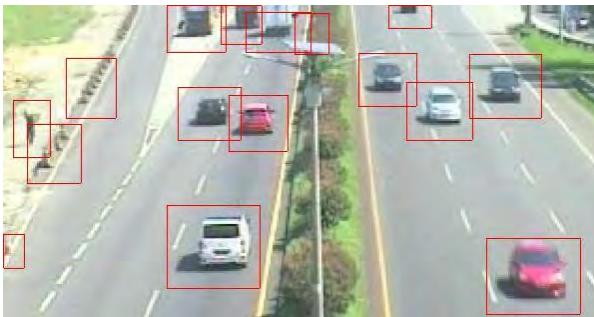
Pada CCTV Cipularang, nilai rata-rata *recall* dari *layer* ini adalah 94% serta untuk nilai rata-rata *precision* adalah 49.1%. Terdapat total 59 mobil terdeteksi dengan benar, namun tidak

mendeteksi 5 mobil. Selain itu ada 60 *tag* dengan yang bukan mobil. Contoh hasil *CCTV* Cipularang dapat dilihat pada Gambar 5.2.



**Gambar 5.2 Hasil Pengujian pada Citra *CCTV* Cipularang Menggunakan Fuzzy Morphology**

Pada *CCTV* Padalarang, nilai rata-rata *recall* dari *layer* ini adalah 91.5% serta untuk nilai rata-rata *precision* adalah 40.85%. Terdapat total 54 mobil terdeteksi dengan benar, namun tidak mendeteksi 5 mobil. Selain itu ada 74 *tag* dengan yang bukan mobil. Contoh hasil *CCTV* Padalarang dapat dilihat pada Gambar 5.3.

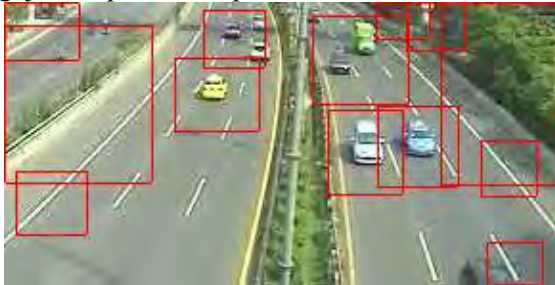


**Gambar 5.3 Hasil Pengujian pada Citra *CCTV* Padalarang Menggunakan Fuzzy Morphology**

### 5.5. Skenario Pengujian 2: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra banyak CCTV menggunakan Fuzzy Morphology.

Pada skenario ini dilakukan uji coba sistem deteksi mobil pada banyak CCTV menggunakan *Fuzzy Morphology* dengan menggunakan parameter yang sama. Dataset yang digunakan didapat dari berbagai macam CCTV. Uji coba menggunakan 47 citra CCTV dan dilakukan perhitungan nilai *recall* dan nilai *precision* dan juga lama proses.

Hasil uji coba ini dapat dilihat pada lampiran. Tampak dari 47 citra masukan, nilai rata-rata *recall* dari *layer* ini adalah 53.9% serta untuk nilai rata-rata *precision* adalah 32.4%. Terdapat total 155 mobil terdeteksi dengan benar, namun tidak mendeteksi 101 mobil. Selain itu ada 280 *tag* dengan yang bukan mobil. Contoh hasil pengujian dapat dilihat pada Gambar 5.4.



**Gambar 5.4 Pengujian pada Citra Banyak CCTV Menggunakan Fuzzy Morphology**

### 5.6. Skenario Pengujian 3: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra satu CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel RGB.

Pada skenario ini dilakukan uji coba sistem deteksi mobil menggunakan *Fuzzy Morphology*, *Gradient Boosting* dan ekstraksi

fitur pada *channel RGB*. Uji coba menggunakan 3 *CCTV*, yaitu *CCTV Padalarang*, *CCTV Semarang* dan *CCTV Cipularang*, tiap tiap *CCTV* terdapat 10 citra, ekstraksi fitur dilakukan pada *channel RGB*, dataset yang digunakan berdasarkan *CCTV* yang digunakan. dan dilakukan perhitungan nilai *recall* dan nilai *precision* dihitung berdasarkan ketentuan berikut:

1. *True positif*: sebuah *tag* dengan objek mobil.
2. *False negatif*: sebuah mobil namun tanpa *tag*.
3. *False positif*: sebuah *tag* namun tanpa objek mobil.

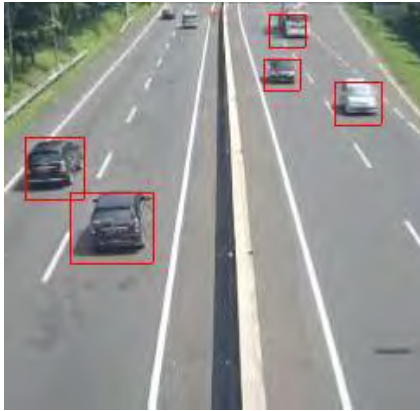
Pada *CCTV Semarang*, skenario ini menggunakan median filter dengan ukuran *structuring element* yaitu 1, high pass filter, Fuzzy dilation dengan *structuring element* yaitu 4, *otsu threshold*, *structuring element* pada operasi *opening* dan erosi yaitu 2, dan ekstraksi fitur pada *channel RGB* sehingga menghasilkan 54 fitur. Pada *CCTV Cipularang*, skenario ini menggunakan citra masukan yang nilai intensitasnya telah dinaikkan sebanyak 50 tiap pixel, nilai *threshold* yaitu 1, Pada *CCTV Padalarang*, skenario ini menggunakan citra masukan yang lebar ukurannya diperbesar menjadi 600 pixel dan tinggi ukuran mengikuti, *otsu threshold*.

Hasil uji coba ini dapat dilihat pada lampiran. Pada *CCTV Semarang*, nilai rata-rata *recall* dari *layer* ini adalah 49% serta untuk nilai rata-rata *precision* adalah 75.8%. Terdapat total 18 mobil terdeteksi dengan benar, namun tidak mendeteksi 15 mobil. Selain itu ada 4 *tag* dengan yang bukan mobil. Contoh hasil *CCTV Semarang* dilihat pada Gambar 5.5.

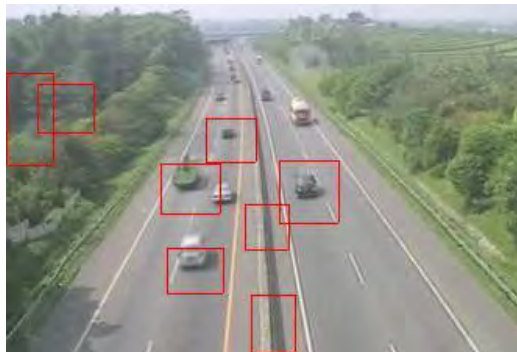
Pada *CCTV Cipularang*, nilai rata-rata *recall* dari *layer* ini adalah 73.3% serta untuk nilai rata-rata *precision* adalah 48.8%. Terdapat total 45 mobil terdeteksi dengan benar, namun tidak mendeteksi 14 mobil. Selain itu ada 23 *tag* dengan yang bukan mobil. Contoh hasil *CCTV Cipularang* dapat dilihat pada Gambar 5.6.

Pada *CCTV Padalarang*, nilai rata-rata *recall* dari *layer* ini adalah 68.1% serta untuk nilai rata-rata *precision* adalah 49.8%. Terdapat total 38 mobil terdeteksi dengan benar, namun tidak

mendeteksi 12 mobil. Selain itu ada 31 *tag* dengan yang bukan mobil. Contoh hasil *CCTV* Padalarang dapat dilihat pada Gambar 5.7.



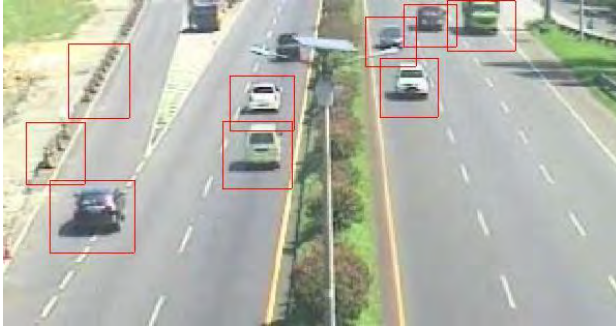
**Gambar 5.5 Hasil Pengujian pada Citra *CCTV* Semarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB**



**Gambar 5.6 Hasil Pengujian pada Citra *CCTV* Cipularang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB**

Dari Hasil uji coba pada ketiga *CCTV*, nilai rata-rata *recall* adalah 63.4% serta untuk nilai rata-rata *precision* adalah 58.1%.

Terdapat total 104 mobil terdeteksi dengan benar, namun tidak mendeteksi 43 mobil. Selain itu ada 86 *tag* dengan yang bukan mobil.



**Gambar 5.7 Hasil Pengujian pada Citra CCTV Padalarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB**

#### **5.7. Skenario Pengujian 4: Perhitungan nilai recall, precision berdasarkan pengujian pada citra banyak CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel RGB.**

Pada skenario ini dilakukan uji coba sistem deteksi mobil pada banyak CCTV dengan menggunakan parameter yang sama. Dataset yang digunakan didapat dari berbagai macam CCTV. Uji coba menggunakan 47 citra CCTV dan dilakukan perhitungan nilai *recall* dan nilai *precision* dan juga lama proses. Pada tahap ini dilakukan ekstraksi fitur pada channel RGB.

Hasil uji coba ini dapat dilihat pada lampiran. Tampak dari 47 citra masukan, nilai rata-rata *recall* dari *layer* ini adalah 29.3% serta untuk nilai rata-rata *precision* adalah 35.6%. Terdapat total 71 mobil terdeteksi dengan benar, namun tidak mendeteksi 143

mobil. Selain itu ada 127 *tag* dengan yang bukan mobil. Contoh hasil pengujian dapat dilihat pada Gambar 5.8



**Gambar 5.8 Hasil Pengujian pada Citra Banyak CCTV Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel RGB**

### **5.8. Skenario Pengujian 5: Perhitungan nilai recall, precision berdasarkan pengujian pada citra satu CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel Grayscale.**

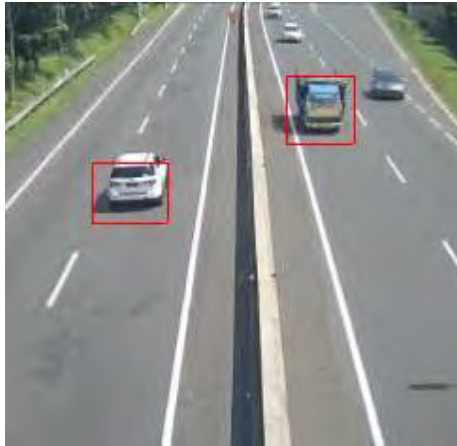
Pada skenario ini dilakukan uji coba sistem deteksi mobil menggunakan fitur pada channel grayscale.. Uji coba menggunakan tiga CCTV tiap tiap CCTV terdapat 10 citra, dan dilakukan perhitungan nilai *recall* dan nilai *precision*. Nilai tersebut dihitung berdasarkan ketentuan berikut:

1. *True positif*: sebuah *tag* dengan objek mobil.
2. *False negatif*: sebuah mobil namun tanpa *tag*.
3. *False positif*: sebuah *tag* namun tanpa objek mobil.

Pada CCTV Semarang, skenario ini menggunakan median filter dengan ukuran *structuring element* yaitu 1, high pass filter, Fuzzy dilation dengan *structuring element* yaitu 4, *otsu threshold*, *structuring element* pada operasi *opening* dan erosi yaitu 2, dan

ekstraksi fitur pada channel RGB sehingga menghasilkan 54 fitur. Pada *CCTV* Cipularang, skenario ini menggunakan citra masukan yang nilai intensitasnya telah dinaikkan sebanyak 50 tiap pixel, nilai *threshold* yaitu 1, Pada *CCTV* Padalarang, skenario ini menggunakan citra masukan yang lebar ukurannya diperbesar menjadi 600 pixel dan tinggi ukuran mengikuti *otsu threshold*.

Hasil uji coba ini dapat dilihat pada lampiran. Pada *CCTV* Semarang, nilai rata-rata *recall* dari *layer* ini adalah 38.35% serta untuk nilai rata-rata *precision* adalah 100%. Terdapat total 16 mobil terdeteksi dengan benar, namun tidak mendeteksi 24 mobil dan tidak ada *tag* dengan yang bukan mobil. Contoh hasil *CCTV* Semarang dilihat pada Gambar 5.9.

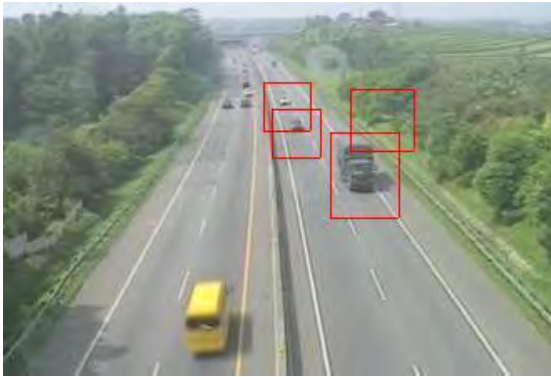


**Gambar 5.9 Hasil Pengujian pada Citra *CCTV* Semarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale**

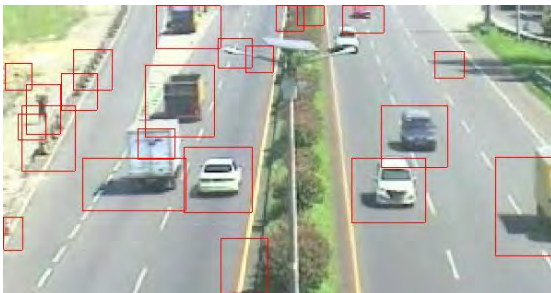
Pada *CCTV* Cipularang, nilai rata-rata *recall* dari *layer* ini adalah 76.2% serta untuk nilai rata-rata *precision* adalah 65.8%. Terdapat total 45 mobil terdeteksi dengan benar, namun tidak mendeteksi 14 mobil. Selain itu ada 25 *tag* dengan yang bukan mobil. Contoh hasil *CCTV* Cipularang dapat dilihat pada Gambar 5.10



Pada *CCTV* Padalarang, nilai rata-rata *recall* dari *layer* ini adalah 91.8% serta untuk nilai rata-rata *precision* adalah 34.1%. Terdapat total 58 mobil terdeteksi dengan benar, namun tidak mendeteksi 5 mobil. Selain itu ada 109 *tag* dengan yang bukan mobil. Contoh hasil *CCTV* Padalarang dapat dilihat pada Gambar 5.11.



**Gambar 5.10 Hasil Pengujian pada Citra *CCTV* Cipularang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale.**



**Gambar 5.11 Hasil Pengujian pada Citra *CCTV* Padalarang Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale**

Dari Hasil uji coba pada ketiga *CCTV*, nilai rata-rata *recall* adalah 68.8% serta untuk nilai rata-rata *precision* adalah 66.6%.

Terdapat total 119 mobil terdeteksi dengan benar, namun tidak mendeteksi 43 mobil. Selain itu ada 134 *tag* dengan yang bukan mobil.

### 5.9. Skenario Pengujian 6: Perhitungan nilai *recall*, *precision* berdasarkan pengujian pada citra banyak CCTV menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel Grayscale

Pada skenario ini dilakukan uji coba sistem deteksi mobil pada banyak CCTV dengan menggunakan parameter yang sama. Dataset yang digunakan didapat dari berbagai macam CCTV. Uji coba menggunakan 47 citra CCTV dan dilakukan perhitungan nilai *recall* dan nilai *precision* dan juga lama proses. Pada pengujian ini dilakukan ekstraksi fitur pada *channel Grayscale*.

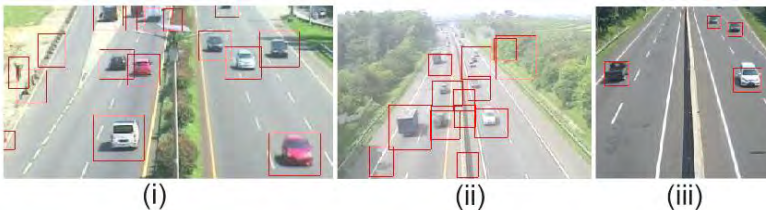
Hasil uji coba ini dapat dilihat pada lampiran. Tampak dari 47 citra masukan, nilai rata-rata *recall* dari *layer* ini adalah 26.3% serta untuk nilai rata-rata *precision* adalah 33.7%. Terdapat total 63 mobil terdeteksi dengan benar, namun tidak mendeteksi 156 mobil. Selain itu ada 79 *tag* dengan yang bukan mobil. Contoh hasil pengujian dapat dilihat pada Gambar 5.12



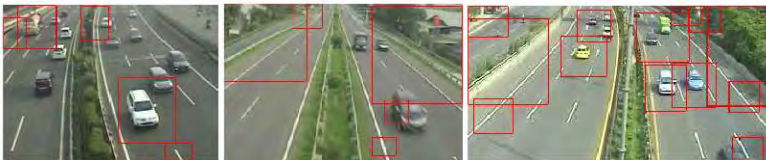
**Gambar 5.12 Hasil Pengujian pada Citra Banyak CCTV Menggunakan Fuzzy Morphology, Gradient Bosting dan Ekstraksi Fitur Pada Channel Grayscale**

### 5.10. Evaluasi

Dari skenario uji coba satu dan dua hasil penghitungan mobil menggunakan *fuzzy morphology* pada satu CCTV menghasilkan penghitungan lebih tinggi yaitu 91% dibandingkan dengan banyak CCTV sebesar 53.9%, hal ini dikarenakan parameter yang digunakan pada satu CCTV disesuaikan pada CCTV yang digunakan, pada banyak CCTV parameter yang digunakan sama untuk berbagai CCTV. Hasil skenario uji coba 1 dan 2 dapat dilihat pada Gambar 5.13 dan Gambar 5.14 dan perbandingan nilai recall dan precision dapat dilihat pada Tabel 5.1.



**Gambar 5.13 Hasil Uji Coba Menggunakan Fuzzy Morphology pada CCTV Padalarang (i), CCTV Cipularang (ii), dan CCTV Semarang (iii)**



**Gambar 5.14 Hasil Uji Coba Menggunakan Fuzzy Morphology pada Banyak CCTV dengan Parameter yang Sama**

**Tabel 5.1 Perbandingan Nilai Recall dan Precision Hasil Fuzzy Morphology antara Satu CCTV dengan Banyak CCTV**

Rata-rata	Recall	Precision
Satu CCTV	0.53991	0.32462
Banyak CCTV	0.91057	0.60164

Dari skenario uji coba tiga dan lima, hasil penghitungan mobil pada satu CCTV Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel Grayscale menghasilkan penghitungan yang lebih tinggi dibandingkan menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel RGB. Perbandingan nilai recall dan precision dapat dilihat pada Tabel 5.2.

**Tabel 5.2 Perbandingan Nilai Recall dan Precision pada Penghitungan Mobil pada Satu CCTV Menggunakan Ekstraksi Fitur Channel Grayscale dengan Channel RGB**

Satu CCTV	Recall	Precision
Ekstraksi fitur pada channel RGB	0.634934	0.581943
Ekstraksi fitur pada channel Grayscale	0.688135	0.666639

Dari skenario uji coba empat dan enam, hasil penghitungan mobil pada banyak CCTV Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel RGB menghasilkan penghitungan yang lebih tinggi dibandingkan menggunakan Fuzzy Morphology, Gradient Bosting dan ekstraksi fitur pada channel Grayscale. Perbandingan nilai recall dan precision dapat dilihat pada Tabel 5.3.

**Tabel 5.3 Perbandingan Nilai Recall dan Precision pada Penghitungan Mobil pada Banyak CCTV Menggunakan Ekstraksi Fitur Channel Grayscale dengan Channel RGB**

Banyak CCTV	Recall	Precision
Ekstraksi fitur pada channel RGB	0.293444	0.356696
Ekstraksi fitur pada channel Grayscale	0.260337	0.337856

Terdapat beberapa hal yang menjadi evaluasi pada sistem penghitungan mobil ini, yaitu kualitas citra yang berbeda beda sehingga sulitnya untuk menentukan operasi operasi apa saja yang digunakan supaya bisa diterapkan pada banyak CCTV. Selain itu

sulitnya menentukan model dataset yang digunakan menjadikan nilai *recall* dan *precision* menjadi rendah.

Pada skenario pengujian yang ada, skala mobil pada gambar sangat berpengaruh, apabila skala mobil kecil dan berdekatan akan susah untuk dideteksi dan cenderung dideteksi satu mobil. Hal ini disebabkan ukuran *structuring element* yang terlalu besar dan hasil saturation yang memiliki intensitas yang hampir sama, contoh dapat dilihat pada Gambar 5.15. Akurasi operasi fuzzy morphology sangat bergantung pada operasi *enhancement* yang digunakan, apabila operasi *enhancement* yang digunakan kurang tepat, akurasi sangat rendah.



**Gambar 5.15 Citra CCTV dengan Kendaraan yang Berdekatan**

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN

### Lampiran A. 1 Hasil Uji Coba Deteksi Mobil pada Citra Satu CCTV Menggunakan Fuzzy Morphology, Gradient Boosting dan Ekstraksi Fitur pada Channel RGB

<b>CCTV CIPULARANG</b>					
<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
0.jpg	7	1	6	0.875	0.538461538
1.jpg	5	1	9	0.833333	0.357142857
2.jpg	7	1	9	0.875	0.4375
3.jpg	2	1	6	0.666667	0.25
4.jpg	4	3	4	0.571429	0.5
5.jpg	6	1	4	0.857143	0.6
6.jpg	5	1	6	0.833333	0.454545455
7.jpg	1	1	3	0.5	0.25
8.jpg	4	5	1	0.444444	0.8
9.jpg	7	1	3	0.875	0.7
<b>TOTAL</b>	<b>48</b>	<b>16</b>	<b>51</b>		
<b>Rata-rata</b>				<b>0.733135</b>	<b>0.488764985</b>

<b>CCTV PADALARANG</b>					
<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
0.jpg	4	2	2	0.666667	0.666667
1.jpg	4	2	3	0.666667	0.571429
2.jpg	7	1	3	0.875	0.7
3.jpg	3	0	3	1	0.5
4.jpg	7	1	2	0.875	0.777778
5.jpg	4	1	3	0.8	0.571429
6.jpg	1	1	4	0.5	0.2

CCTV PADALARANG					
NAMA FILE	TP	FN	FP	Recall	Precision
7.jpg	5	1	5	0.833333	0.5
8.jpg	3	2	3	0.6	0.5
9.jpg	0	1	3	0	0
TOTAL	38	12	31		
Rata-rata				0.681667	0.49873

CCTV SEMARANG					
NAMA FILE	TP	FN	FP	Recall	Precision
0.jpg	2	1	1	0.666667	0.666667
1.jpg	3	2	0	0.6	1
2.jpg	3	1	0	0.75	1
3.jpg	1	1	0	0.5	1
4.jpg	1	2	0	0.333333	1
5.jpg	3	1	1	0.75	0.75
6.jpg	2	3	0	0.4	1
7.jpg	0	0	0	0	0
8.jpg	2	3	1	0.4	0.666667
9.jpg	1	1	1	0.5	0.5
TOTAL	18	15	4		
Rata-rata				0.49	0.758333

	TP	FN	FP
CCTV CIPULARANG	48	16	51
CCTV PADALARANG	38	12	31
CCTV SEMARANG	18	15	4
TOTAL	104	43	86



	<b>Recall</b>	<b>Precision</b>
<b>CCTV CIPULARANG</b>	0.733135	0.488765
<b>CCTV PADALARANG</b>	0.681667	0.49873
<b>CCTV SEMARANG</b>	0.49	0.758333
<b>Rata-rata</b>	<b>0.634934</b>	<b>0.581943</b>

**Lampiran A. 2 Hasil Uji Coba Deteksi Mobil pada Citra Banyak CCTV Menggunakan Fuzzy Morphology, Gradient Boosting Dan Ekstraksi Fitur Pada Channel RGB**

<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
1.jpg	4	6	1	0.4	0.8
2.jpg	1	2	6	0.333333	0.142857
3.jpg	4	1	6	0.8	0.4
4.jpg	2	2	2	0.5	0.5
5.jpg	6	1	5	0.857143	0.545455
6.jpg	2	2	2	0.5	0.5
7.jpg	1	6	1	0.142857	0.5
8.jpg	1	6	0	0.142857	1
9.jpg	3	7	1	0.3	0.75
10.jpg	6	1	1	0.857143	0.857143
11.jpg	0	12	1	0	0
12.jpg	3	7	2	0.3	0.6
13.jpg	1	4	0	0.2	1
14.jpg	2	3	0	0.4	1
15.jpg	0	4	5	0	0
16.jpg	0	3	4	0	0
17.jpg	1	1	2	0.5	0.333333
18.jpg	1	4	2	0.2	0.333333
19.jpg	0	1	3	0	0

<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
20.jpg	0	1	3	0	0
21.jpg	0	3	1	0	0
22.jpg	1	4	1	0.2	0.5
23.jpg	0	1	3	0	0
24.jpg	0	1	5	0	0
25.jpg	0	1	4	0	0
26.jpg	1	0	3	1	0.25
27.jpg	0	1	1	0	0
28.jpg	0	2	3	0	0
29.jpg	0	1	3	0	0
30.jpg	1	1	1	0.5	0.5
31.jpg	0	1	4	0	0
32.jpg	1	0	2	1	0.333333
33.jpg	1	3	0	0.25	1
34.jpg	0	1	5	0	0
35.jpg	0	1	2	0	0
36.jpg	2	4	4	0.333333	0.333333
37.jpg	2	4	3	0.333333	0.4
38.jpg	6	5	5	0.545455	0.545455
39.jpg	1	3	4	0.25	0.2
40.jpg	2	1	5	0.666667	0.285714
41.jpg	1	3	3	0.25	0.25
42.jpg	2	5	6	0.285714	0.25
43.jpg	4	2	3	0.666667	0.571429
44.jpg	2	5	2	0.285714	0.5
45.jpg	1	5	5	0.166667	0.166667
46.jpg	2	6	1	0.25	0.666667
47.jpg	3	5	1	0.375	0.75

NAMA FILE	TP	FN	FP	Recall	Precision
<b>TOTAL</b>	<b>71</b>	<b>143</b>	<b>127</b>		
<b>Rata-rata</b>				<b>0.293444</b>	<b>0.356696</b>

**Lampiran A. 3 Hasil Uji Coba Deteksi Mobil pada Citra Satu CCTV Menggunakan Fuzzy Morphology, Gradient Boosting dan Ekstraksi Fitur pada Channel Grayscale**

<b>CCTV CIPULARANG</b>					
NAMA FILE	TP	FN	FP	Recall	Precision
0.jpg	6	3	7	0.666667	0.461538
1.jpg	3	1	1	0.75	0.75
2.jpg	6	0	3	1	0.666667
3.jpg	3	0	4	1	0.428571
4.jpg	3	4	4	0.428571	0.428571
5.jpg	7	1	2	0.875	0.777778
6.jpg	5	1	0	0.833333	1
7.jpg	1	1	1	0.5	0.5
8.jpg	5	2	2	0.714286	0.714286
9.jpg	6	1	1	0.857143	0.857143
<b>TOTAL</b>	<b>45</b>	<b>14</b>	<b>25</b>		
<b>Rata-rata</b>				<b>0.7625</b>	<b>0.658455</b>

<b>CCTV PADALARANG</b>					
NAMA FILE	TP	FN	FP	Recall	Precision
0.jpg	7	0	10	1	0.411765
1.jpg	7	1	14	0.875	0.333333
2.jpg	10	0	9	1	0.526316
3.jpg	4	0	15	1	0.210526

<b>CCTV PADALARANG</b>					
<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
4.jpg	7	1	11	0.875	0.388889
5.jpg	5	1	9	0.833333	0.357143
6.jpg	3	0	9	1	0.25
7.jpg	6	0	11	1	0.352941
8.jpg	6	0	11	1	0.352941
9.jpg	3	2	10	0.6	0.230769
<b>TOTAL</b>	<b>58</b>	<b>5</b>	<b>109</b>		
<b>Rata-rata</b>				<b>0.918333</b>	<b>0.341462</b>

<b>CCTV SEMARANG</b>					
<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
0.jpg	0	3	0	0	1
1.jpg	4	1	0	0.8	1
2.jpg	2	2	0	0.5	1
3.jpg	1	0	0	1	1
4.jpg	1	3	0	0.25	1
5.jpg	3	2	0	0.6	1
6.jpg	2	5	0	0.285714	1
7.jpg	1	2	0	0	1
8.jpg	2	3	0	0.4	1
9.jpg	0	3	0	0	1
<b>TOTAL</b>	<b>16</b>	<b>24</b>	<b>0</b>		
<b>Rata-rata</b>				<b>0.383571</b>	<b>1</b>

	<b>TP</b>	<b>FN</b>	<b>FP</b>
<b>CCTV CIPULARANG</b>	45	14	25
<b>CCTV PADALARANG</b>	58	5	109
<b>CCTV SEMARANG</b>	16	24	0
<b>Rata-rata</b>	<b>119</b>	<b>43</b>	<b>134</b>

	<b>Recall</b>	<b>Precision</b>
<b>CCTV CIPULARANG</b>	0.7625	0.658455
<b>CCTV PADALARANG</b>	0.918333	0.341462
<b>CCTV SEMARANG</b>	0.383571	1
<b>Rata-rata</b>	<b>0.688135</b>	<b>0.666639</b>

**Lampiran A. 4 Hasil Uji Coba Deteksi Mobil pada Citra Banyak CCTV Menggunakan Fuzzy Morphology, Gradient Boosting dan Ekstraksi Fitur pada Channel Grayscale**

<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
1.jpg	4	6	1	0.4	0.8
2.jpg	2	3	3	0.4	0.4
3.jpg	3	4	8	0.428571	0.272727
4.jpg	2	5	3	0.285714	0.4
5.jpg	4	4	7	0.5	0.363636
6.jpg	2	2	4	0.5	0.333333
7.jpg	0	8	0	0	0
8.jpg	3	4	0	0.428571	1
9.jpg	3	5	1	0.375	0.75
10.jpg	2	6	0	0.25	1
11.jpg	2	10	3	0.166667	0.4
12.jpg	3	8	2	0.272727	0.6
13.jpg	1	4	0	0.2	1

<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
14.jpg	2	3	1	0.4	0.666667
15.jpg	0	4	1	0	0
16.jpg	1	2	2	0.333333	0.333333
17.jpg	1	2	1	0.333333	0.5
18.jpg	3	2	0	0.6	1
19.jpg	1	0	5	1	0.166667
20.jpg	0	1	1	0	0
21.jpg	0	3	1	0	0
22.jpg	0	4	3	0	0
23.jpg	0	1	0	0	0
24.jpg	0	1	2	0	0
25.jpg	0	1	3	0	0
26.jpg	1	1	1	0.5	0
27.jpg	0	1	0	0	0
28.jpg	0	2	0	0	0
29.jpg	0	1	2	0	0
30.jpg	1	1	0	0.5	1
31.jpg	0	1	0	0	0
32.jpg	1	0	1	1	0.5
33.jpg	1	3	1	0.25	0.5
34.jpg	0	1	1	0	0
35.jpg	0	1	0	0	0
36.jpg	0	5	3	0	0
37.jpg	0	6	0	0	0
38.jpg	0	9	0	0	0
39.jpg	1	3	1	0.25	0.5
40.jpg	0	3	2	0	0
41.jpg	1	2	4	0.333333	0.2

NAMA FILE	TP	FN	FP	Recall	Precision
42.jpg	8	0	2	1	0.8
43.jpg	2	3	2	0.4	0.5
44.jpg	3	4	1	0.428571	0.75
45.jpg	1	4	6	0.2	0.142857
46.jpg	0	8	0	0	0
47.jpg	4	4	0	0.5	1
<b>TOTAL</b>	<b>63</b>	<b>156</b>	<b>79</b>		
<b>Rata-rata</b>				<b>0.260337</b>	<b>0.337856</b>

**Lampiran A. 5 Hasil Uji Coba Deteksi Mobil pada Citra  
Banyak CCTV Menggunakan Fuzzy Morphology**

NAMA FILE	TP	FN	FP	Recall	Precision
1.jpg	8	6	1	0.571429	0.888889
2.jpg	6	1	17	0.857143	0.26087
3.jpg	8	3	10	0.727273	0.444444
4.jpg	6	5	16	0.545455	0.272727
5.jpg	9	2	10	0.818182	0.473684
6.jpg	3	1	6	0.75	0.333333
7.jpg	6	4	2	0.6	0.75
8.jpg	7	6	3	0.538462	0.7
9.jpg	7	3	2	0.7	0.777778
10.jpg	7	2	2	0.777778	0.777778
11.jpg	2	10	3	0.166667	0.4
12.jpg	5	5	2	0.5	0.714286
13.jpg	3	2	2	0.6	0.6
14.jpg	4	1	4	0.8	0.5
15.jpg	3	1	9	0.75	0.25

<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
16.jpg	2	1	10	0.666667	0.166667
17.jpg	2	1	7	0.666667	0.222222
18.jpg	4	1	7	0.8	0.363636
19.jpg	1	0	5	1	0.166667
20.jpg	0	1	5	0	0
21.jpg	1	2	4	0.333333	0.2
22.jpg	1	3	4	0.25	0.2
23.jpg	0	1	4	0	0
24.jpg	0	1	7	0	0
25.jpg	0	1	5	0	0
26.jpg	1	1	4	0.5	0.2
27.jpg	0	1	3	0	0
28.jpg	1	2	4	0.333333	0.2
29.jpg	0	1	4	0	0
30.jpg	1	2	4	0.333333	0.2
31.jpg	0	1	5	0	0
32.jpg	1	0	7	1	0.125
33.jpg	1	2	5	0.333333	0.166667
34.jpg	0	1	6	0	0
35.jpg	1	0	4	1	0.2
36.jpg	4	3	9	0.571429	0.307692
37.jpg	3	2	5	0.6	0.375
38.jpg	7	3	8	0.7	0.466667
39.jpg	3	1	8	0.75	0.272727
40.jpg	3	1	9	0.75	0.25
41.jpg	3	0	11	1	0.214286
42.jpg	5	4	8	0.555556	0.384615
43.jpg	6	0	6	1	0.5



NAMA FILE	TP	FN	FP	Recall	Precision
44.jpg	6	2	4	0.75	0.6
45.jpg	4	3	11	0.571429	0.266667
46.jpg	3	6	4	0.333333	0.428571
47.jpg	7	1	4	0.875	0.636364
<b>TOTAL</b>	<b>155</b>	<b>101</b>	<b>280</b>		
<b>Rata-rata</b>				<b>0.539911</b>	<b>0.324622</b>

**Lampiran A. 6 Hasil Uji Coba Deteksi Mobil pada Citra Satu  
CCTV Menggunakan Fuzzy Morphology**

<b>CCTV PADALARANG</b>					
NAMA FILE	TP	FN	FP	Recall	Precision
0.jpg	6	0	8	1	0.428571
1.jpg	7	1	9	0.875	0.4375
2.jpg	9	0	7	1	0.5625
3.jpg	4	0	8	1	0.333333
4.jpg	7	1	5	0.875	0.583333
5.jpg	4	1	8	0.8	0.333333
6.jpg	2	0	6	1	0.25
7.jpg	6	0	8	1	0.428571
8.jpg	6	0	8	1	0.428571
9.jpg	3	2	7	0.6	0.3
<b>TOTAL</b>	<b>54</b>	<b>5</b>	<b>74</b>		
<b>Rata-rata</b>				<b>0.915</b>	<b>0.408571</b>

<b>CCTV SEMARANG</b>					
NAMA FILE	TP	FN	FP	Recall	Precision
0.jpg	3	0	1	1	0.75
1.jpg	5	0	1	1	0.833333

<b>CCTV SEMARANG</b>					
<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
2.jpg	4	0	0	1	1
3.jpg	1	0	0	1	1
4.jpg	3	1	0	0.75	1
5.jpg	4	1	1	0.8	0.8
6.jpg	5	2	0	0.714286	1
7.jpg	3	0	0	1	1
8.jpg	5	1	0	0.833333	1
9.jpg	2	1	1	0.666667	0.666667
<b>TOTAL</b>	<b>35</b>	<b>6</b>	<b>4</b>		
<b>Rata-rata</b>				<b>0.876429</b>	<b>0.905</b>

<b>CCTV CIPULARANG</b>					
<b>NAMA FILE</b>	<b>TP</b>	<b>FN</b>	<b>FP</b>	<b>Recall</b>	<b>Precision</b>
0.jpg	7	1	8	0.875	0.466666667
1.jpg	6	0	8	1	0.428571429
2.jpg	7	0	9	1	0.4375
3.jpg	3	0	7	1	0.3
4.jpg	6	2	5	0.75	0.545454545
5.jpg	7	0	6	1	0.538461538
6.jpg	6	0	6	1	0.5
7.jpg	2	0	4	1	0.333333333
8.jpg	8	0	3	1	0.727272727
9.jpg	7	2	4	0.777778	0.636363636
<b>TOTAL</b>	<b>59</b>	<b>5</b>	<b>60</b>		
<b>Rata-rata</b>				<b>0.940278</b>	<b>0.491362388</b>

	<b>TP</b>	<b>FN</b>	<b>FP</b>
<b><i>CCTV CIPULARANG</i></b>	59	5	60
<b><i>CCTV PADALARANG</i></b>	54	5	74
<b><i>CCTV SEMARANG</i></b>	35	6	4
<b>Rata-rata</b>	<b>148</b>	<b>16</b>	<b>138</b>

	<b>Recall</b>	<b>Precision</b>
<b><i>CCTV CIPULARANG</i></b>	0.940278	0.491362
<b><i>CCTV PADALARANG</i></b>	0.915	0.408571
<b><i>CCTV SEMARANG</i></b>	0.876429	0.905
<b>Rata-rata</b>	<b>0.910569</b>	<b>0.601645</b>

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

#### **6.1. Kesimpulan**

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Deteksi mobil menggunakan *Fuzzy Morphology* pada satu CCTV menghasilkan pendeteksian lebih tinggi dibanding banyak CCTV yaitu 91% sedangkan banyak CCTV sebesar 53.9% serta tingkat false alarm yang lebih rendah yaitu 60% sedangkan banyak CCTV sebesar 32%.
2. Deteksi mobil pada satu CCTV menggunakan *Fuzzy Morphology* dan *Gradient Boosting* menghasilkan tingkat pendeteksian lebih tinggi dibanding banyak CCTV yaitu 66.1 % sedangkan banyak CCTV 27,6% serta tingkat false alarm yang lebih rendah yaitu 62% sedangkan banyak CCTV 34%.
3. Pada banyak CCTV, menggunakan fitur pada channel RGB menghasilkan pendeteksian lebih tinggi dibanding channel Grayscale yaitu 29% sedangkan channel Grayscale 26%.
4. Pada satu CCTV, menggunakan fitur pada channel Grayscale menghasilkan pendeteksian lebih tinggi dibanding channel RGB yaitu 68% sedangkan channel RGB 63%.
5. Operasi *Fuzzy Morphology* sangat bergantung pada kualitas gambar, hal ini berpengaruh pada operasi dan parameter yang digunakan.
6. Operasi *Gradient Boosting* sangat bergantung pada hasil morphology yang didapat.

## 6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Menggunakan operasi *Morphology* selain *Fuzzy Morphology* untuk memperbaiki hasil deteksi pada banyak CCTV.
2. Menggunakan ekstraksi fitur selain *Haar Wavelet Transform* dan metode normalisasi selain min-max untuk memperbaiki hasil deteksi menggunakan Gradient Boosting
3. Menggunakan *Region of Interest* (ROI) untuk mengurangi daerah bukan mobil yang terdeteksi.
4. Dilakukan ekstraksi jalan raya untuk meningkatkan hasil deteksi pada mobil.

## DAFTAR PUSTAKA

- [1] Nuryadin, Sony., Syarifuddin, "<http://elib.unikom.ac.id/>," 2006. [Online]. Available: <http://elib.unikom.ac.id/download.php?id=6964>. [Accessed 4 June 2016].
- [2] Dardhmonth, "Image Processing in IDL," [dartmouth.edu](http://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html), 16 June 2006. [Online]. Available: [http://northstar-www.dartmouth.edu/doc/idl/html\\_6.2/Filtering\\_an\\_Imagehvr.html](http://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html). [Accessed 4 June 2016].
- [3] I. The MathWorks, "<http://www.mathworks.com/>," The MathWorks, Inc., 2016. [Online]. Available: <http://www.mathworks.com/help/images/morphological-dilation-and-erosion.html?requestedDomain=www.mathworks.com>. [Accessed 4 Juni 2016].
- [4] Lianto, Joko Buliali., Ahmad Saikhu, Silvester Tena, [6] Chastine Fatichah, "A Hybrid Fuzzy Morphology and Connected," *International Journal on Smart Sensing and Intelligent Systems*, p. Nov, 2014.
- [5] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on System, Man, and Cybernetics.*, Vols. SMC-9, p. 62, 1979.
- [6] Perkins, S., A. Walker, E. Wolfart, R. Fisher, "Image Analysys - Connected Component Labelling," School of Informatics The University of Edinburg, 2003. [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm>. [Accessed 4 Juni 2016].
- [7] J. Utama, "Electrocardiogram (ECG) with Noise Reduction Based on Wavelet Using LabVIEW Programming," *Telekontran*, vol. 1, p. 40, 2013.

- [8] Hasanah, Rini Nur, Purnomo Budi Santosa, Dodit Suprianto, "Sistem Pengenalan Wajah Secara Real-Time," *Jurnal EECCIS*, vol. 7, no. 2, p. 179, 2013.
- [9] Spark, "Ensembles - spark.mllib," Spark, [Online]. Available: <http://spark.apache.org/docs/latest/mllib-ensembles.html>. [Accessed 4 Juni 2016].
- [10] Health Science Library, "Searching\_ - \_Recall\_Precision," [Online]. Available: [https://www.creighton.edu/fileadmin/user/HSL/docs/ref/Searching\\_-\\_Recall\\_Precision.pdf](https://www.creighton.edu/fileadmin/user/HSL/docs/ref/Searching_-_Recall_Precision.pdf).



## BIODATA PENULIS



Achmad Saiful, lahir di Surabaya pada tanggal 23 Maret 1994. Lulusan SMAN 15 Surabaya pada tahun 2012 dan sedang menempuh pendidikan di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya angkatan tahun 2012. Pernah menjadi asisten mata kuliah Struktur data selama 2 semester. Terlibat aktif dalam organisasi kemahasiswaan diantaranya Staff Pengembangan Sumber Daya Mahasiswa Himpunan Mahasiswa Teknik Computer-Informatika (HMTC), Staff Pengembangan Sumber Daya Mahasiswa Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf) dan Kepala Departemen Pengembangan Sumber Daya Mahasiswa Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf).

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Komputasi Cerdas dan Visi (KCV) dan memiliki ketertarikan di bidang komputer visi dan juga manajemen SDM. Penulis dapat dihubungi melalui surel: [achmads123@gmail.com](mailto:achmads123@gmail.com).